20501O"E960h0O1

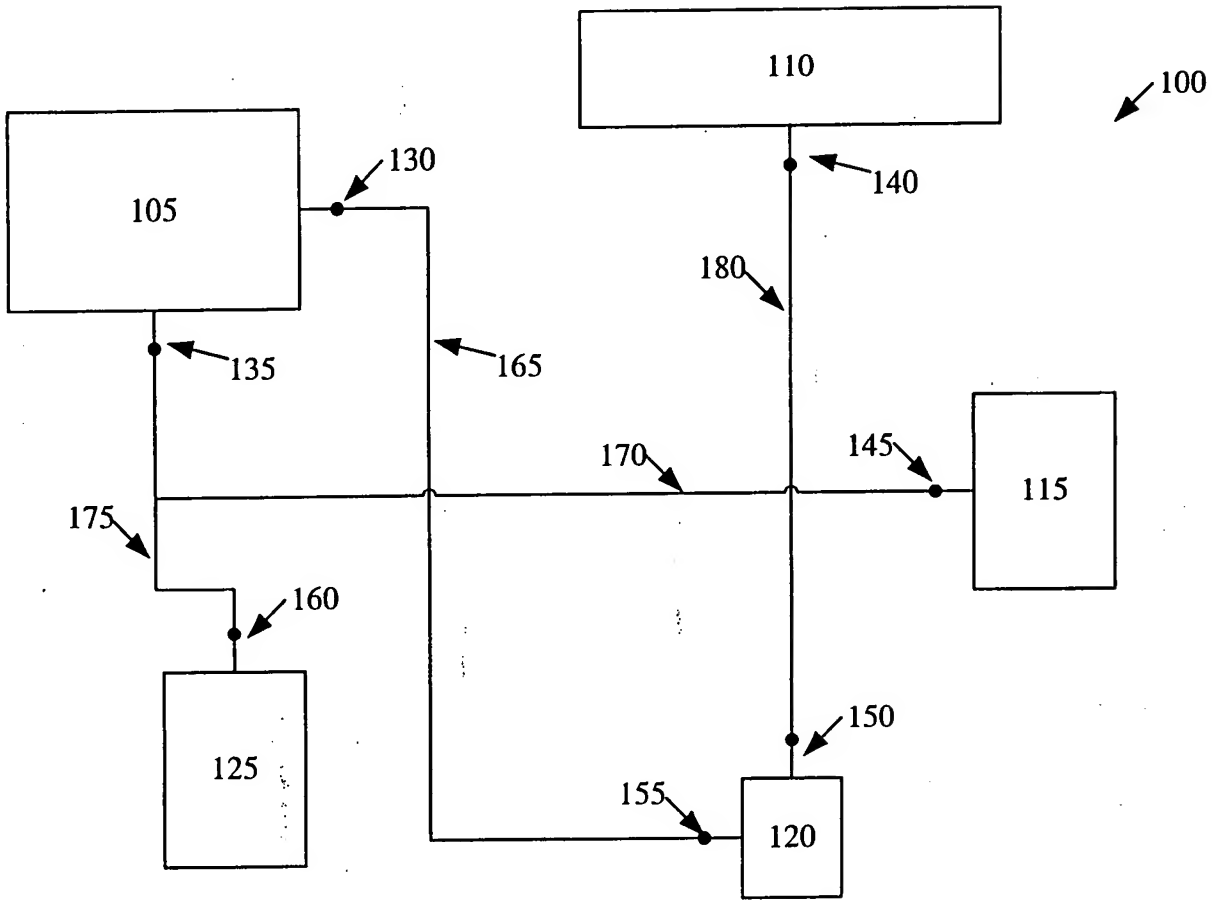

Figure 1

20501O"E960h001

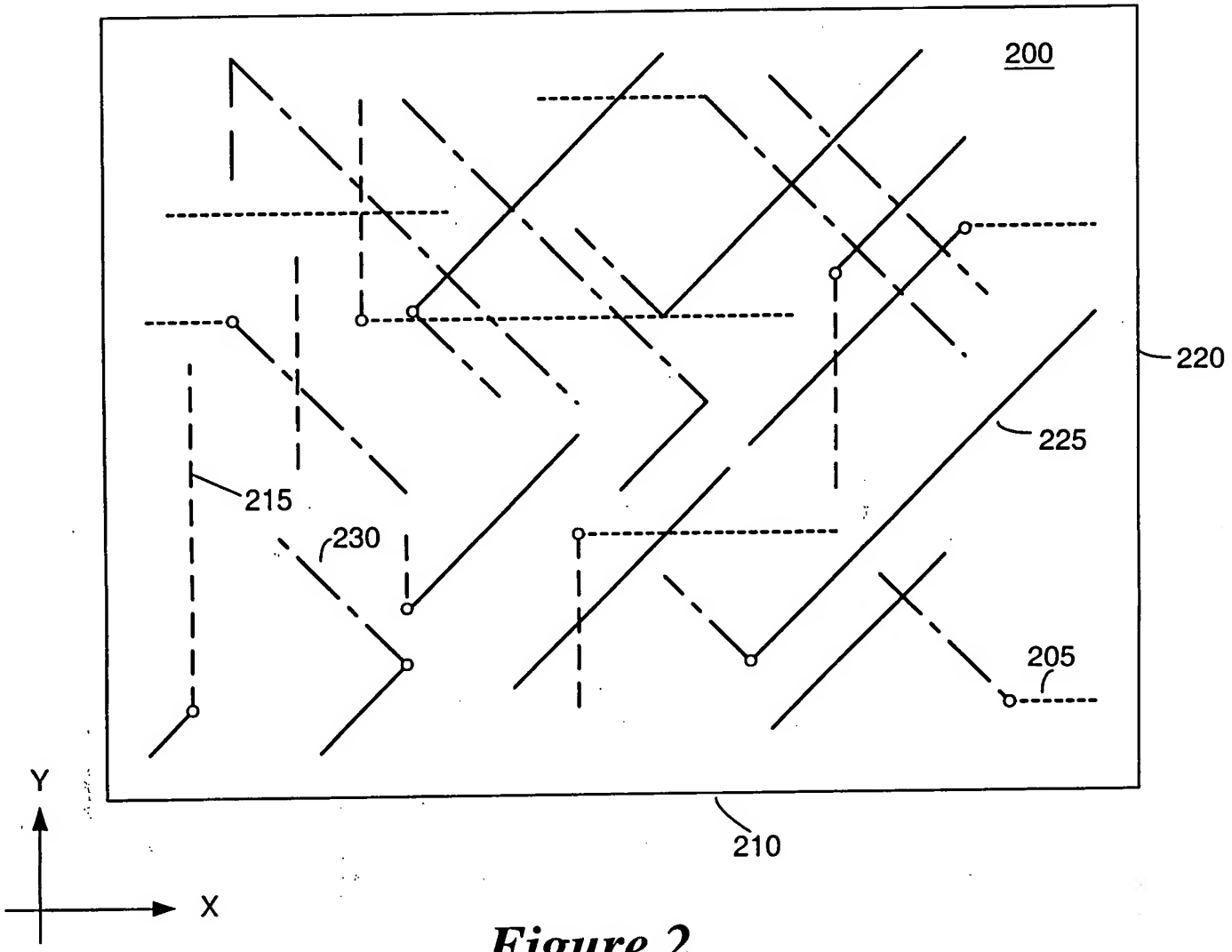

Figure 2

20501D "E960h00T

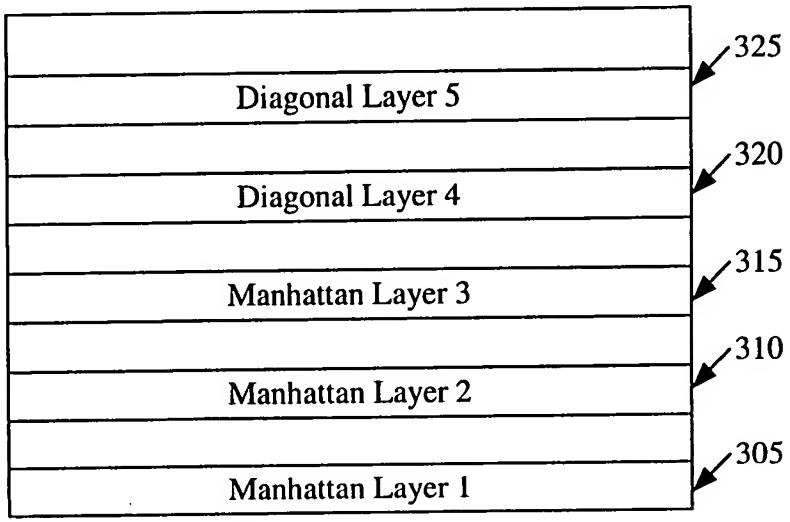

Figure 3

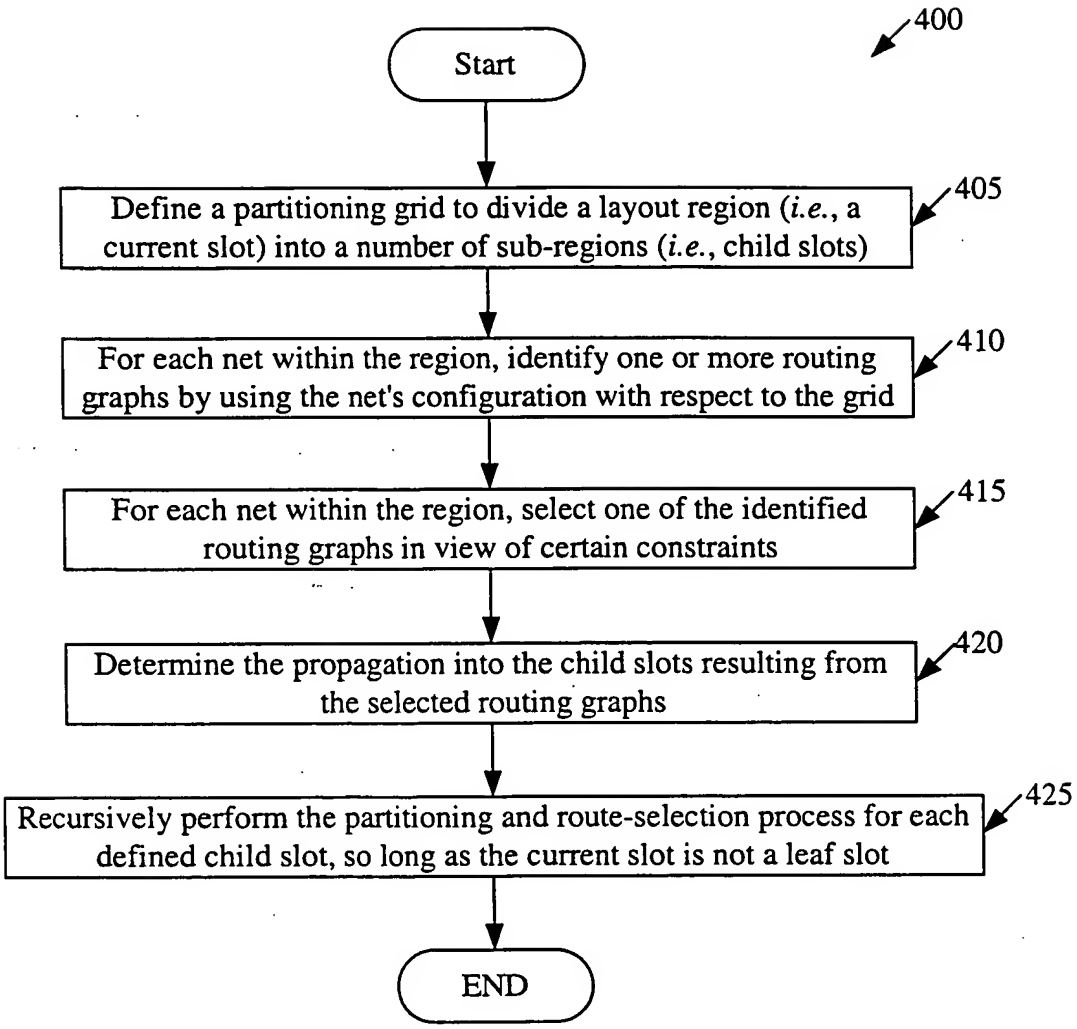
400
Start
Define a partitioning grid to divide a layout region (i.e., a current slot) into a number of sub-regions (i.e., child slots)
405
For each net within the region, identify one or more routing graphs by using the net's configuration with respect to the grid
410
For each net within the region, select one of the identified routing graphs in view of certain constraints
415
Determine the propagation into the child slots resulting from the selected routing graphs
420
Recursively perform the partitioning and route-selection process for each defined child slot, so long as the current slot is not a leaf slot
425
END

Figure 4

20507D "E960h08T

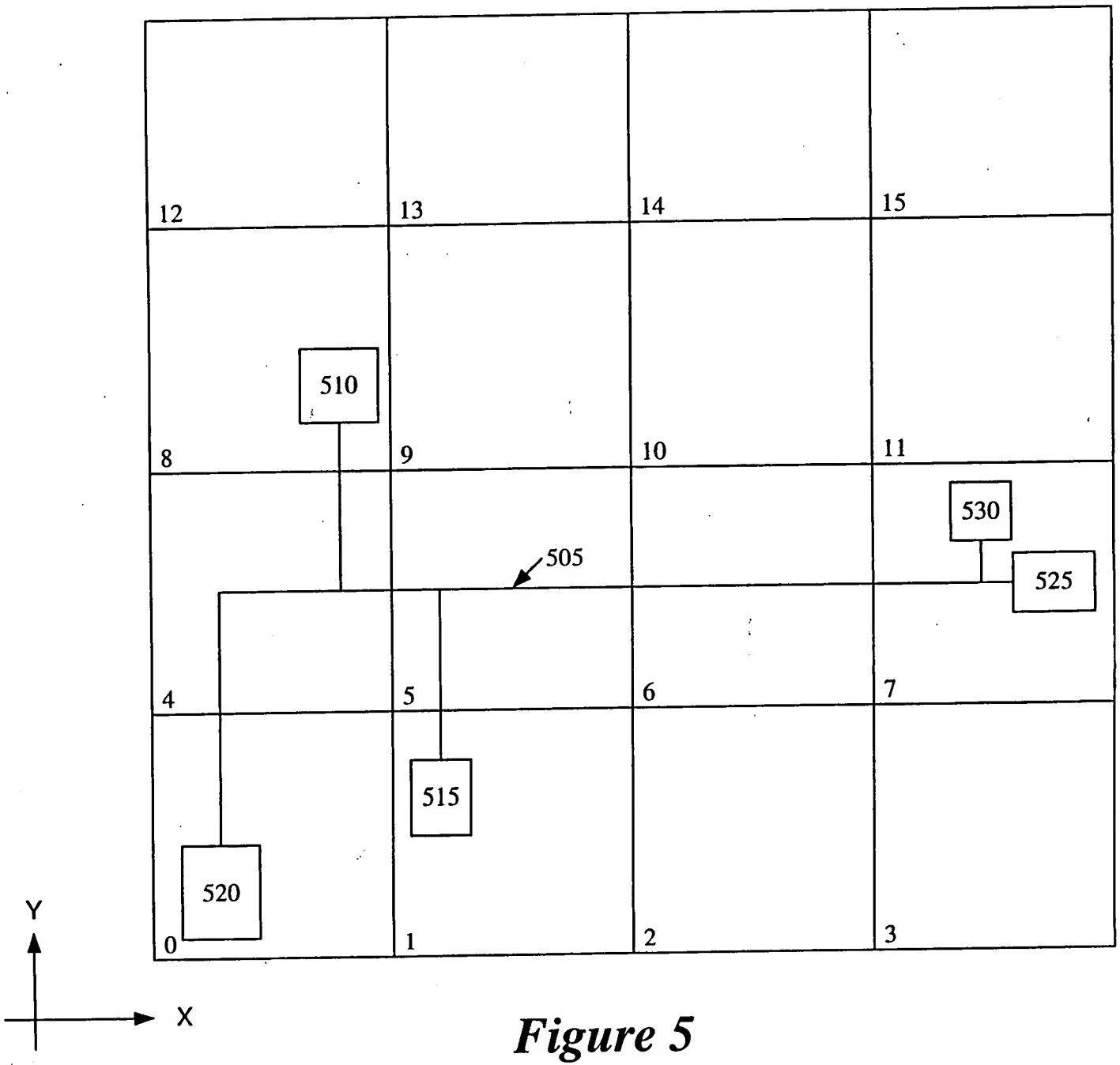

Figure 5

20501D "E960h00T

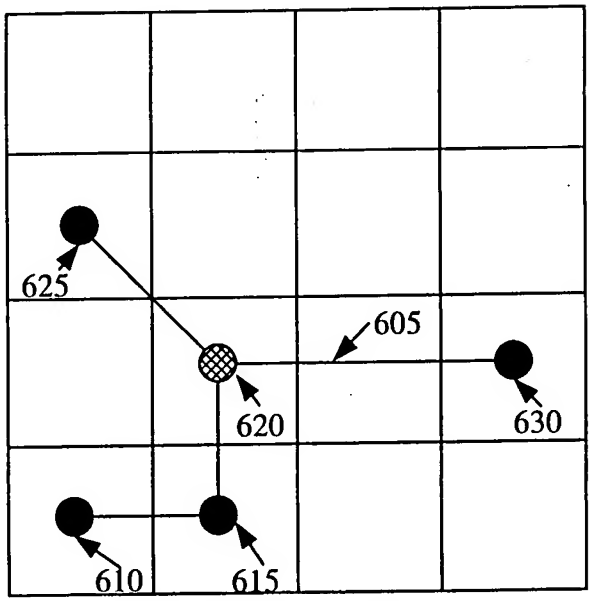

Figure 6

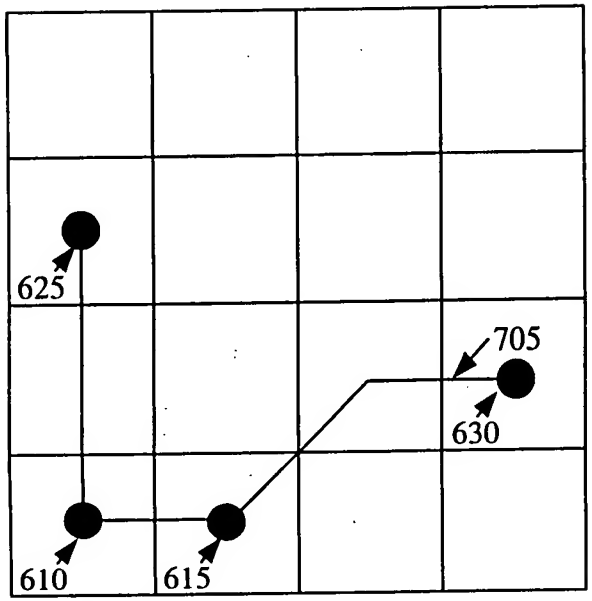

Figure 7

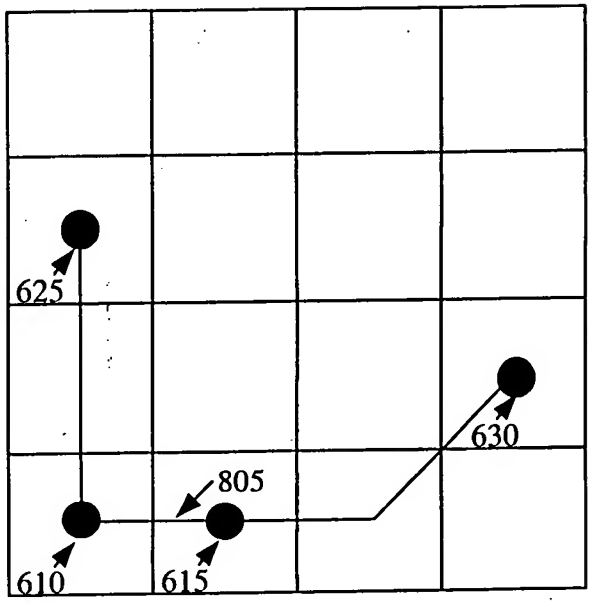

Figure 8

20501D "E960h00T

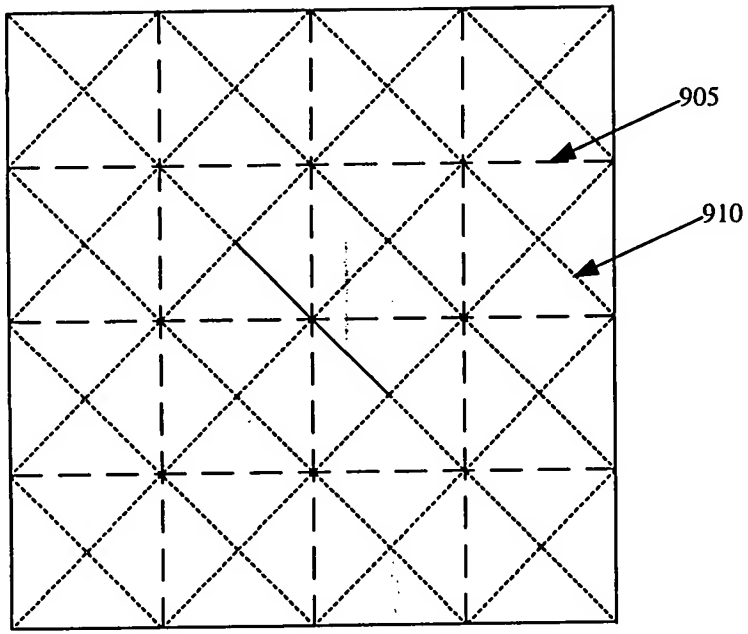

Figure 9

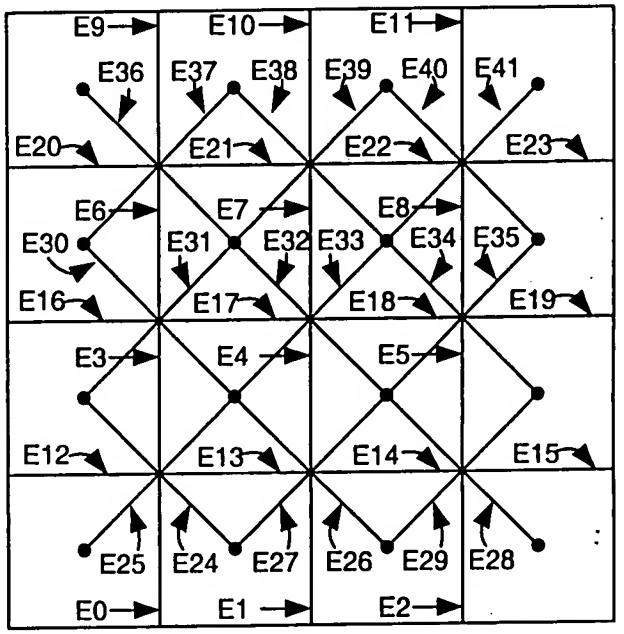

Figure 10

20S0T0"E960h00T

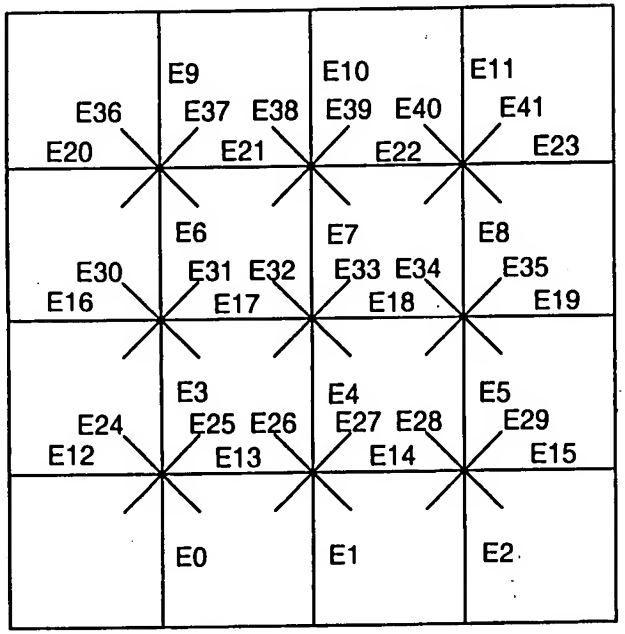

Figure 11

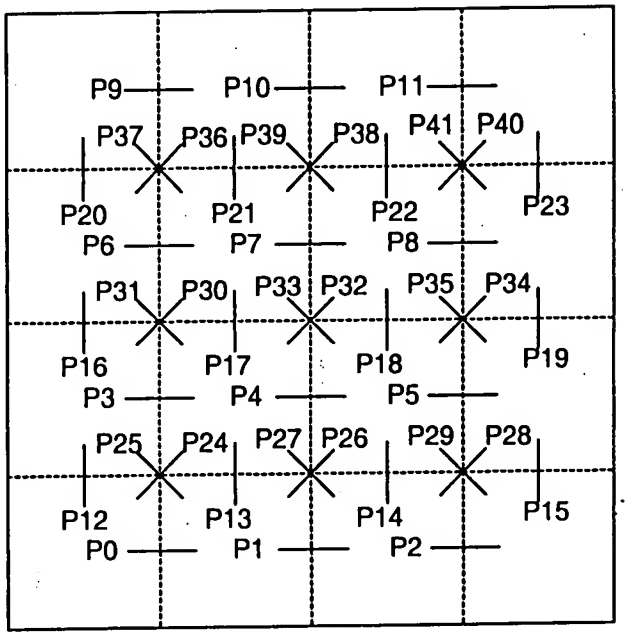

Figure 12

20070 E960h00T

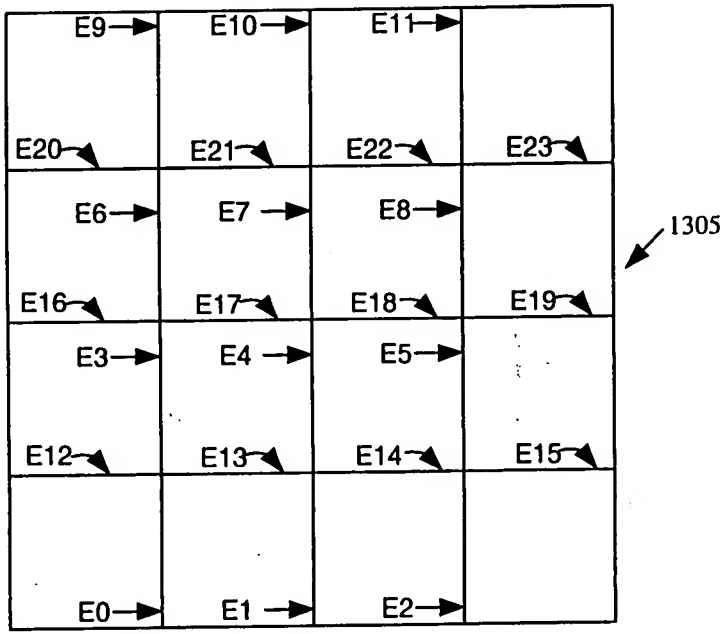

Figure 13

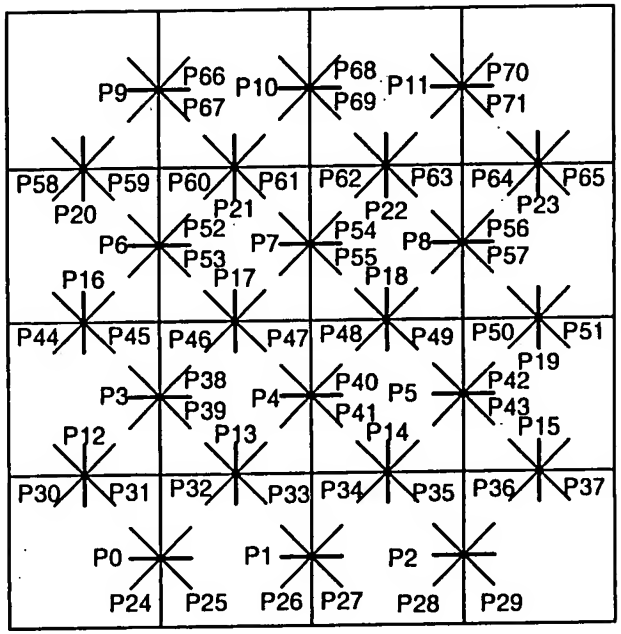

Figure 14

2050TO"E960hDOT

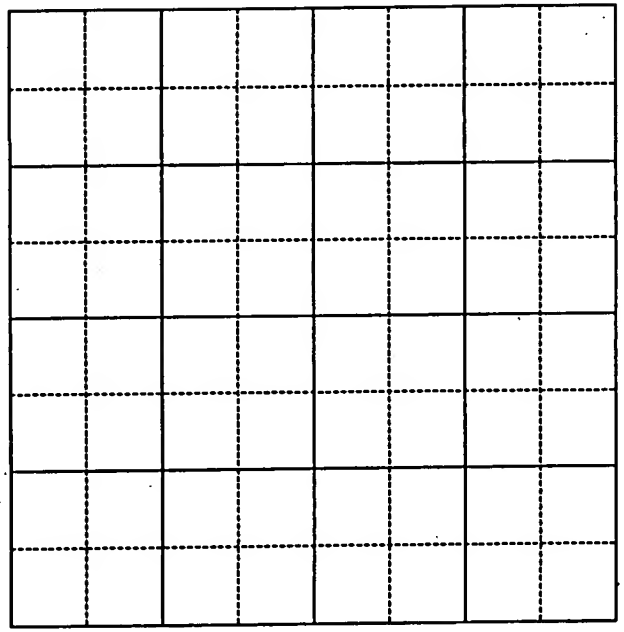

Figure 15

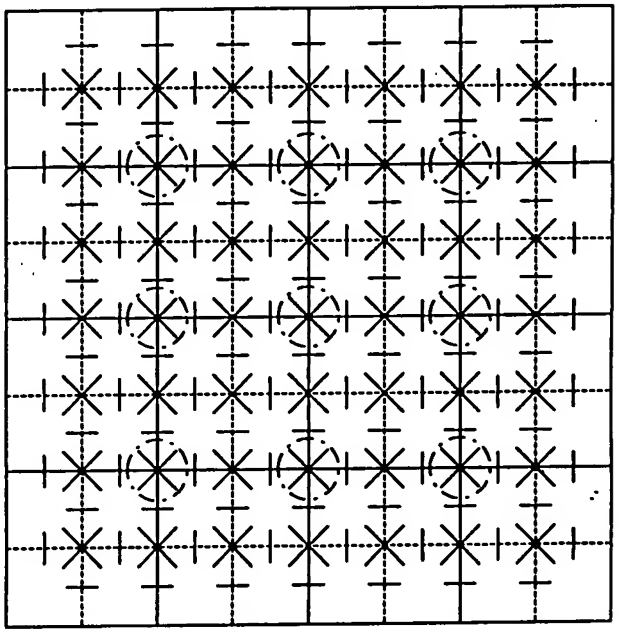

Figure 17

2050TO"E960h00t

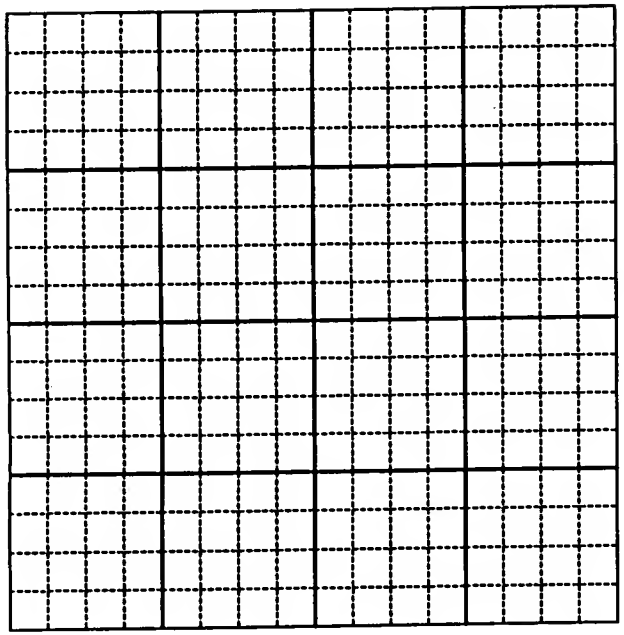

Figure 16

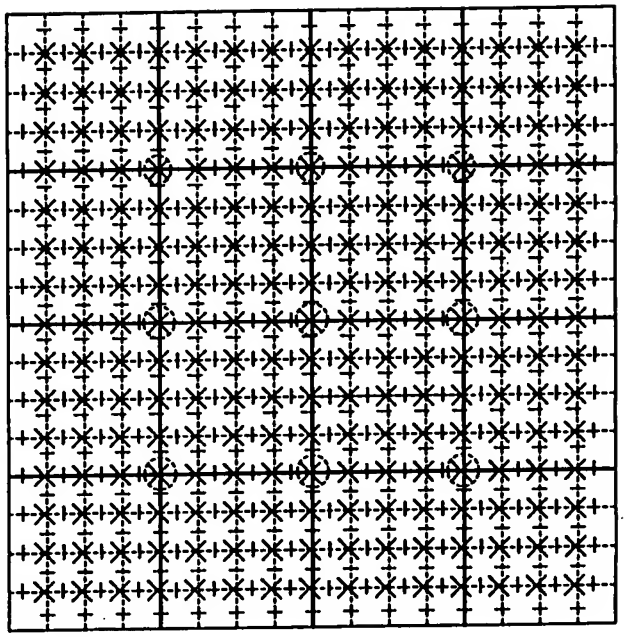

Figure 18

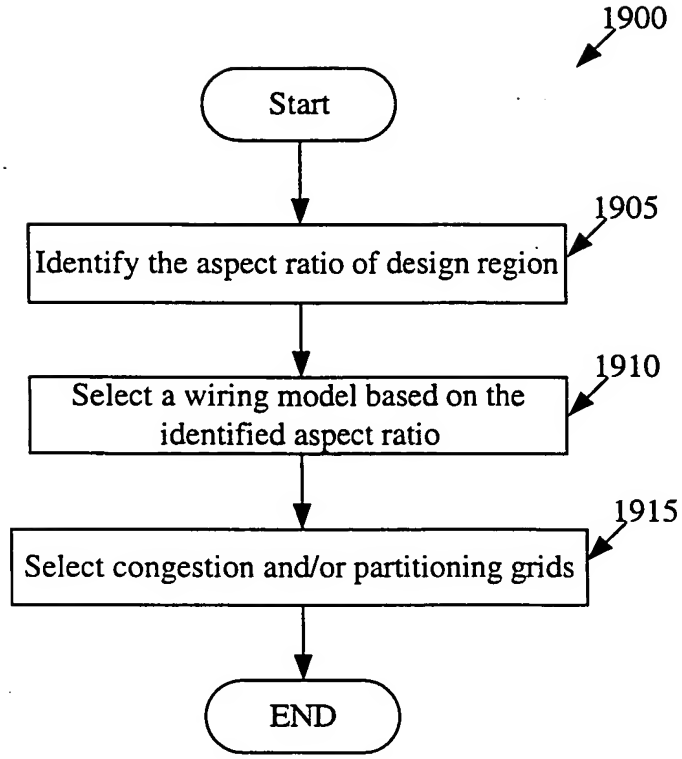
1900
Start
1905
Identify the aspect ratio of design region
1910
Select a wiring model based on the identified aspect ratio
1915
Select congestion and/or partitioning grids
END

Figure 19

20501O" E960h00T

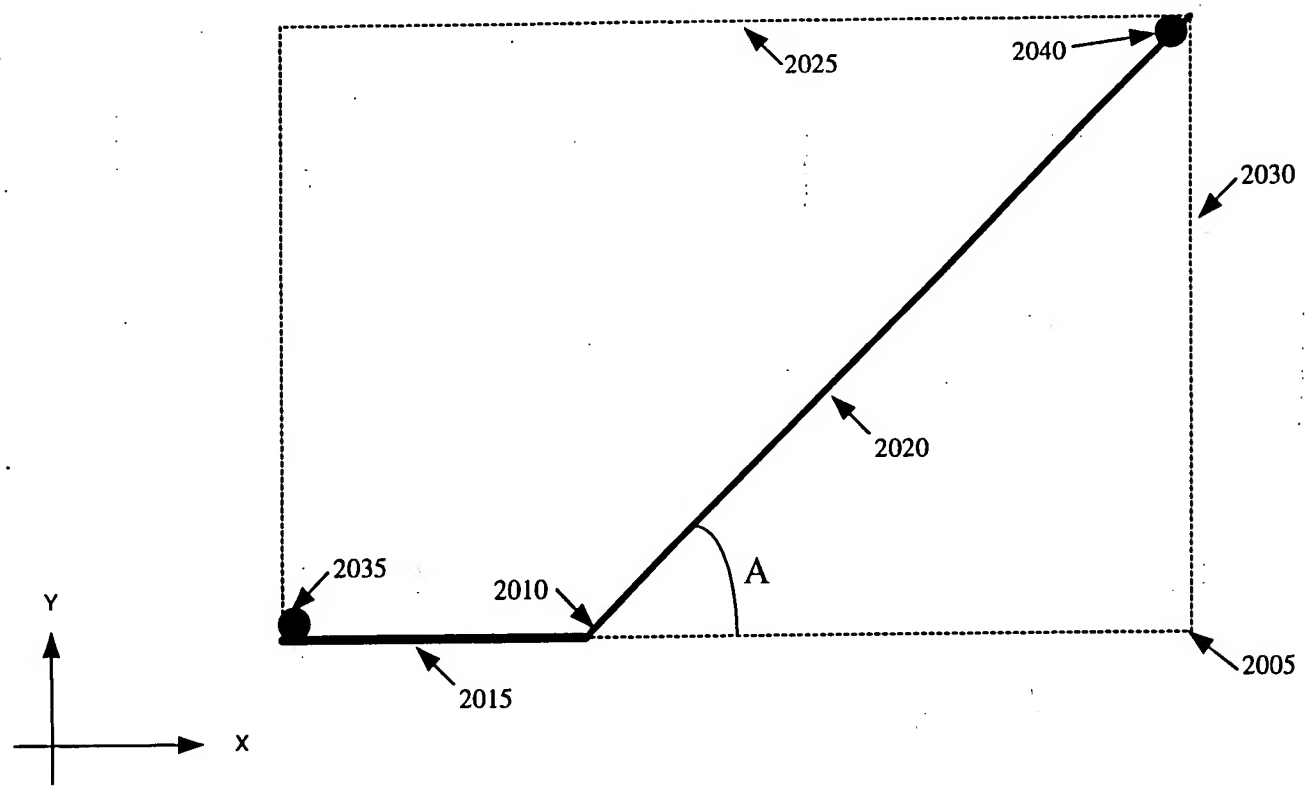

Figure 20

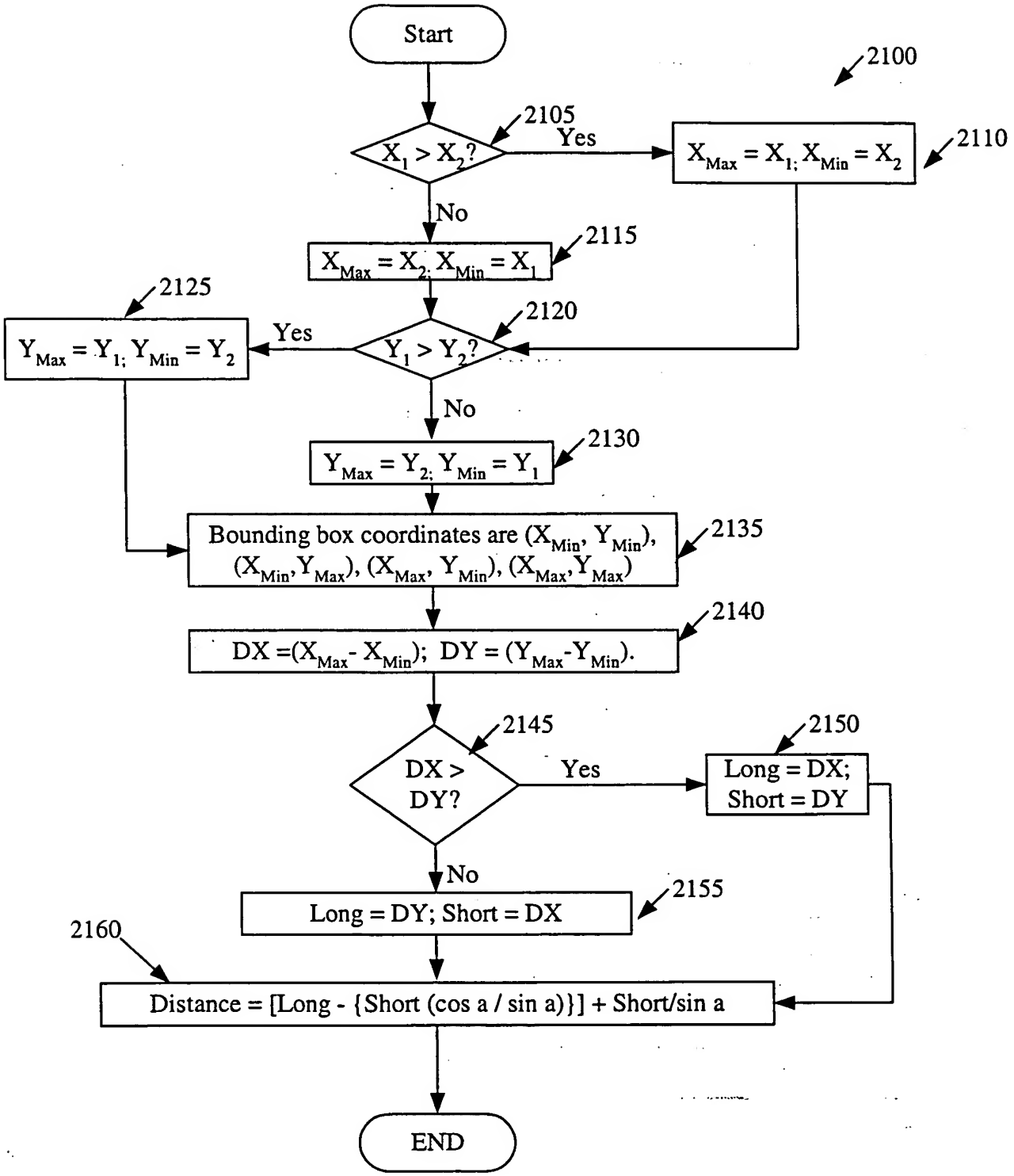
Start
2100
2105
$X_1 > X_2$?
Yes
$X_{Max} = X_1$; $X_{Min} = X_2$
2110
No
$X_{Max} = X_2$; $X_{Min} = X_1$
2115
2125
$Y_{Max} = Y_1$; $Y_{Min} = Y_2$
Yes
2120
$Y_1 > Y_2$?
No
$Y_{Max} = Y_2$; $Y_{Min} = Y_1$
2130
Bounding box coordinates are (X_{Min}, Y_{Min}), (X_{Min},Y_{Max}), (X_{Max}, Y_{Min}), (X_{Max},Y_{Max})
2135
DX =(X_{Max}- X_{Min}); DY = (Y_{Max}-Y_{Min}).
2140
2145
DX > DY?
Yes
2150
Long = DX; Short = DY
No
Long = DY; Short = DX
2155
2160
Distance = [Long - {Short (cos a / sin a)}] + Short/sin a
END

Figure 21

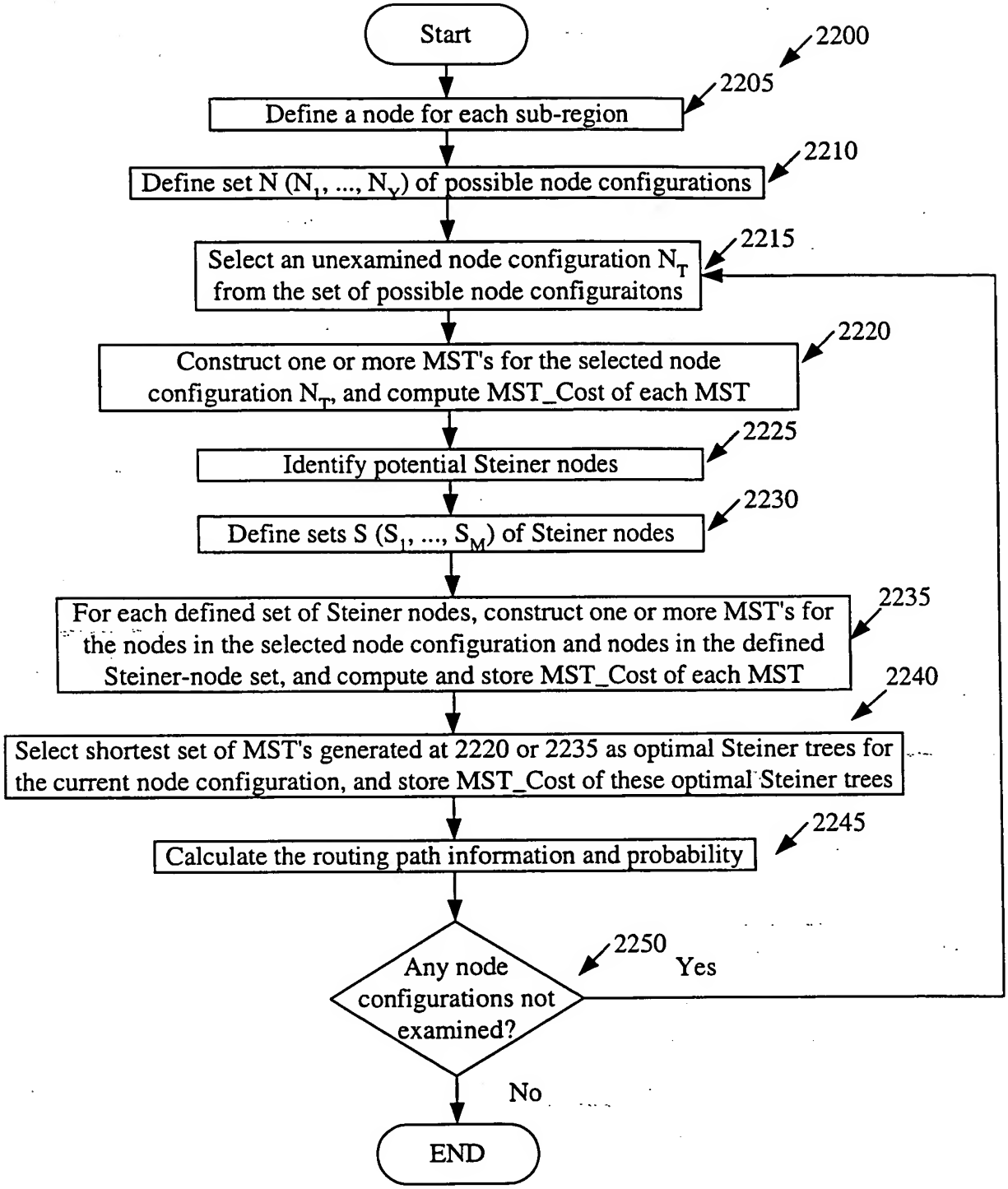
Start
2200
2205
Define a node for each sub-region
2210
Define set N (N_1, ..., N_Y) of possible node configurations
2215
Select an unexamined node configuration N_T from the set of possible node configuraitons
2220
Construct one or more MST's for the selected node configuration N_T, and compute MST_Cost of each MST
2225
Identify potential Steiner nodes
2230
Define sets S (S_1, ..., S_M) of Steiner nodes
2235
For each defined set of Steiner nodes, construct one or more MST's for the nodes in the selected node configuration and nodes in the defined Steiner-node set, and compute and store MST_Cost of each MST
2240
Select shortest set of MST's generated at 2220 or 2235 as optimal Steiner trees for the current node configuration, and store MST_Cost of these optimal Steiner trees
2245
Calculate the routing path information and probability
2250
Any node configurations not examined?
Yes
No
END

Figure 22

20501D "E960h00T

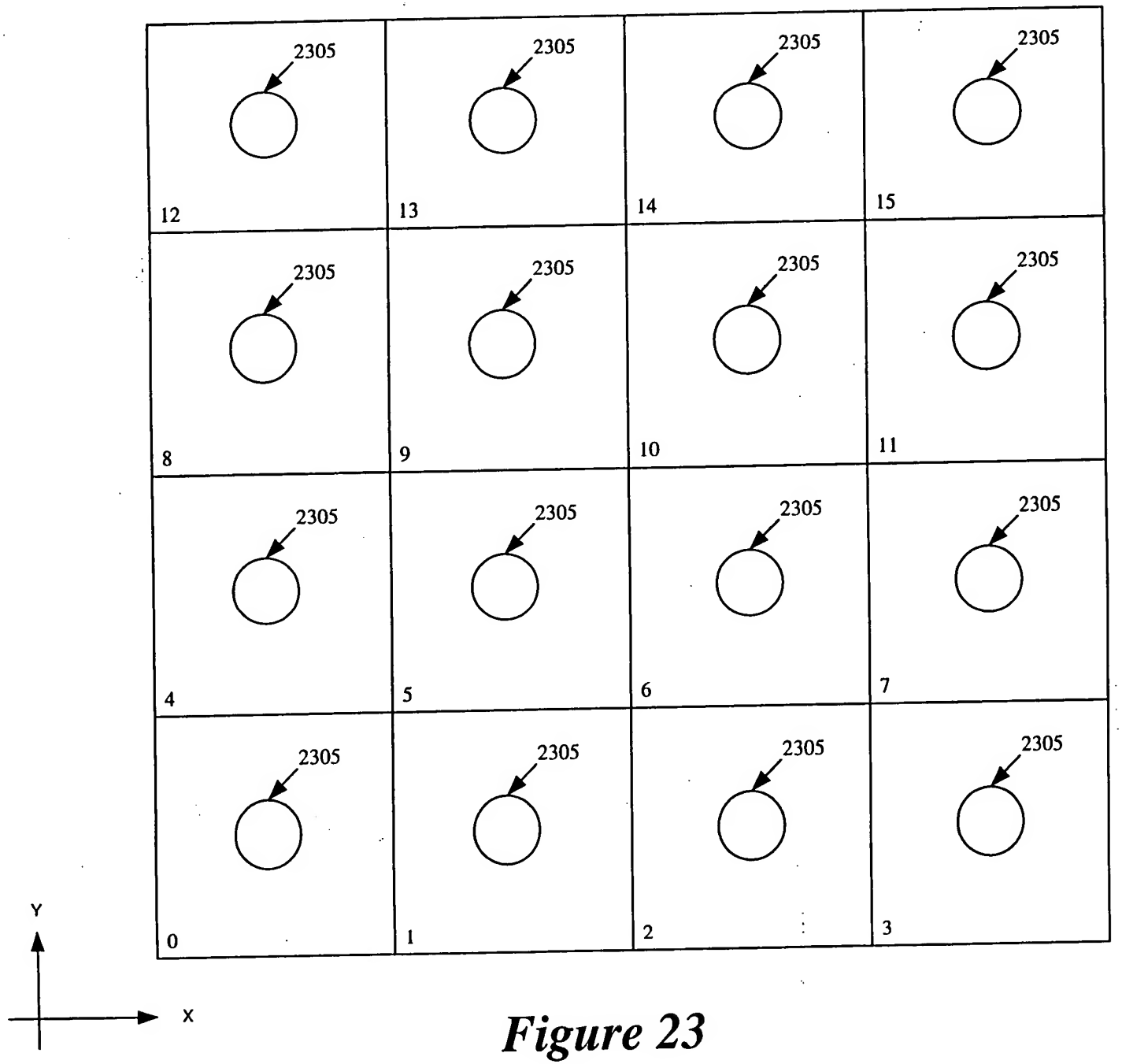

Figure 23

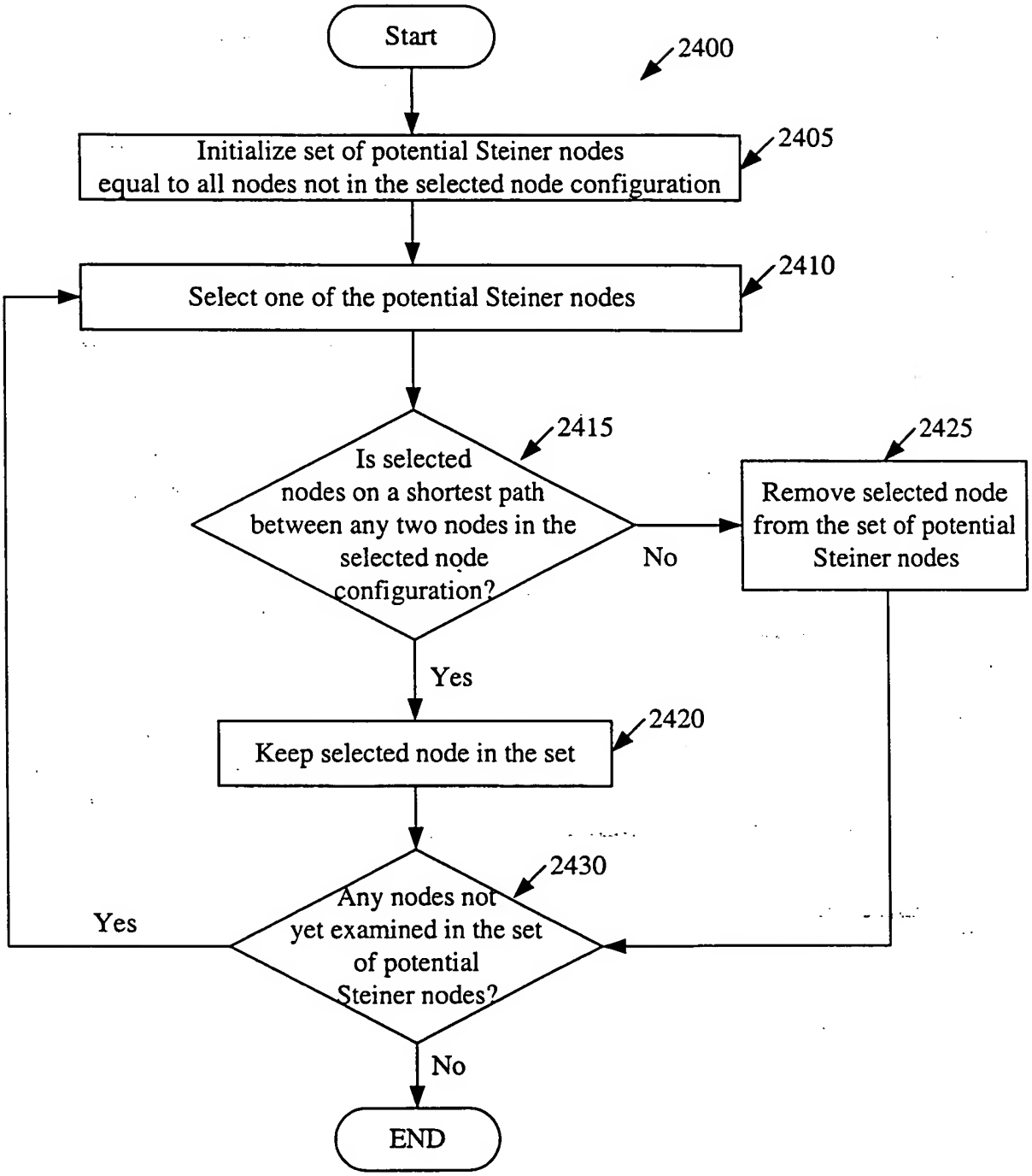
Start
2400
Initialize set of potential Steiner nodes equal to all nodes not in the selected node configuration
2405
Select one of the potential Steiner nodes
2410
Is selected nodes on a shortest path between any two nodes in the selected node configuration?
2415
No
Remove selected node from the set of potential Steiner nodes
2425
Yes
Keep selected node in the set
2420
Any nodes not yet examined in the set of potential Steiner nodes?
2430
Yes
No
END

Figure 24

20050502.3600400

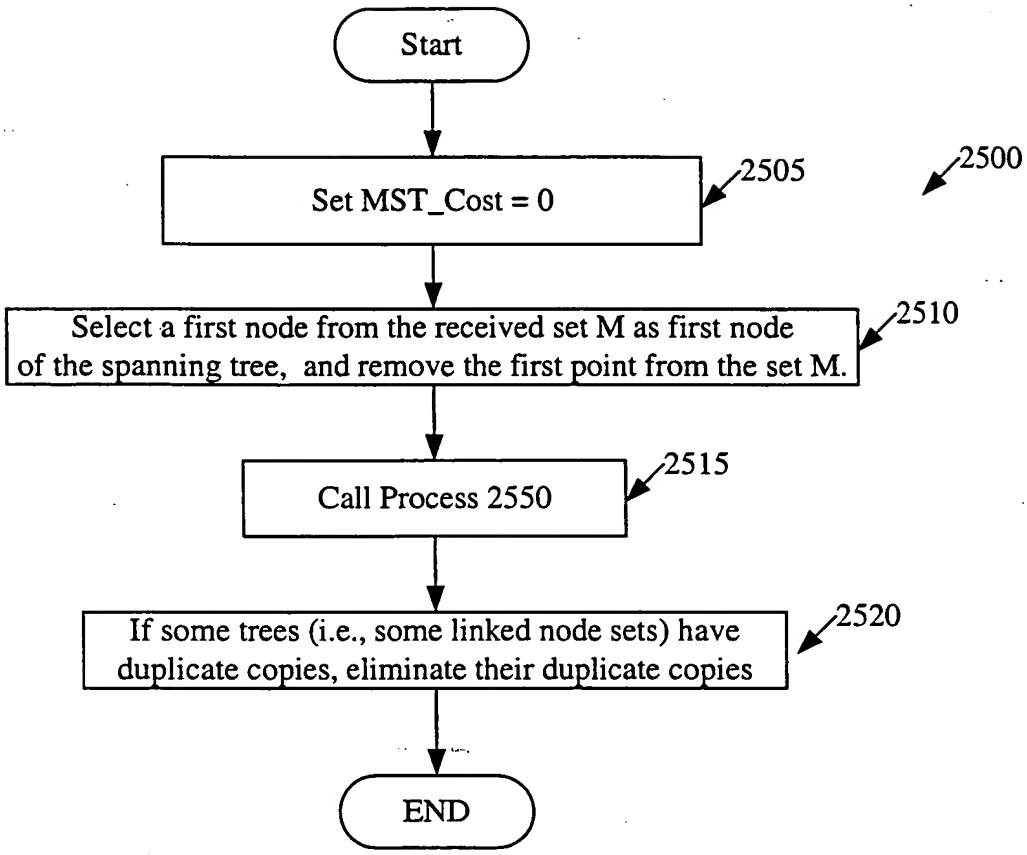

Figure 25A

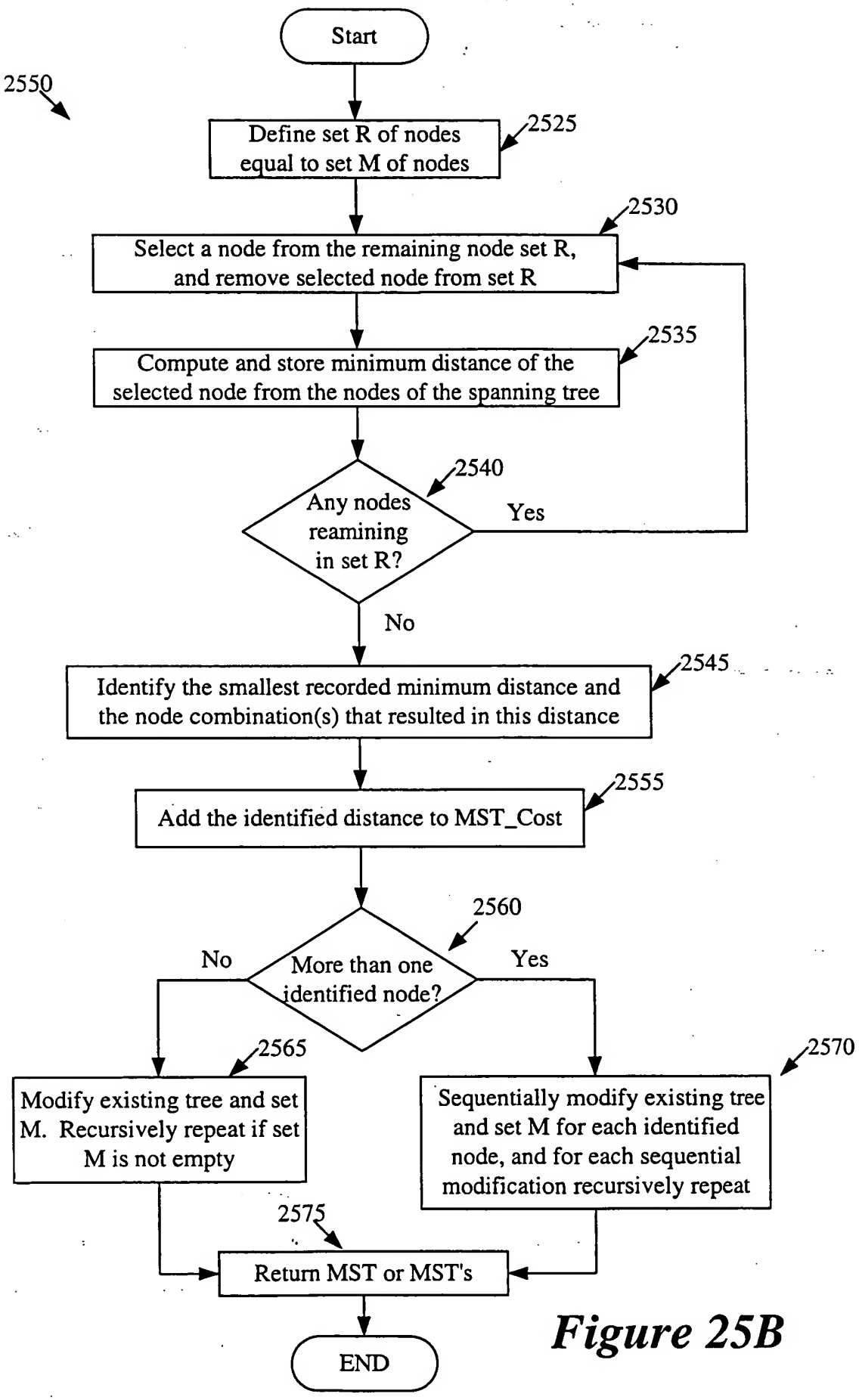
Start
2550
Define set R of nodes equal to set M of nodes
2525
2530
Select a node from the remaining node set R, and remove selected node from set R
2535
Compute and store minimum distance of the selected node from the nodes of the spanning tree
2540
Any nodes reamining in set R?
Yes
No
2545
Identify the smallest recorded minimum distance and the node combination(s) that resulted in this distance
2555
Add the identified distance to MST_Cost
2560
More than one identified node?
No
Yes
2565
Modify existing tree and set M. Recursively repeat if set M is not empty
2570
Sequentially modify existing tree and set M for each identified node, and for each sequential modification recursively repeat
2575
Return MST or MST's
END

Figure 25B

20050243824 - 20051027

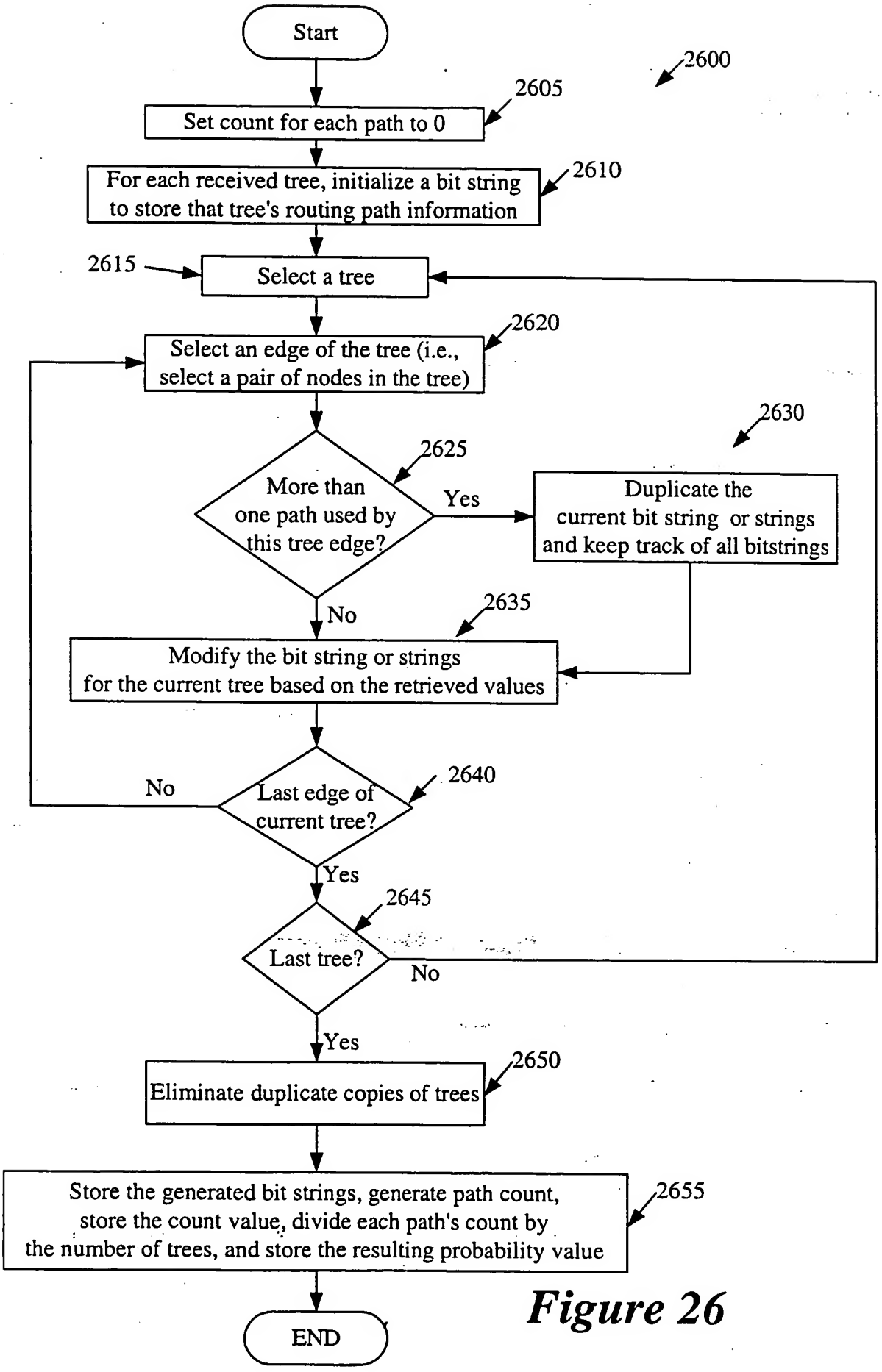

Figure 26

20501O "E960h0T

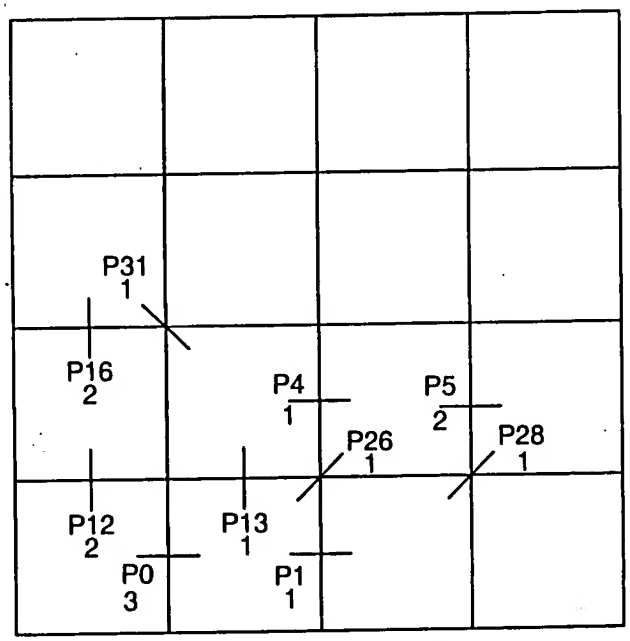

Figure 27

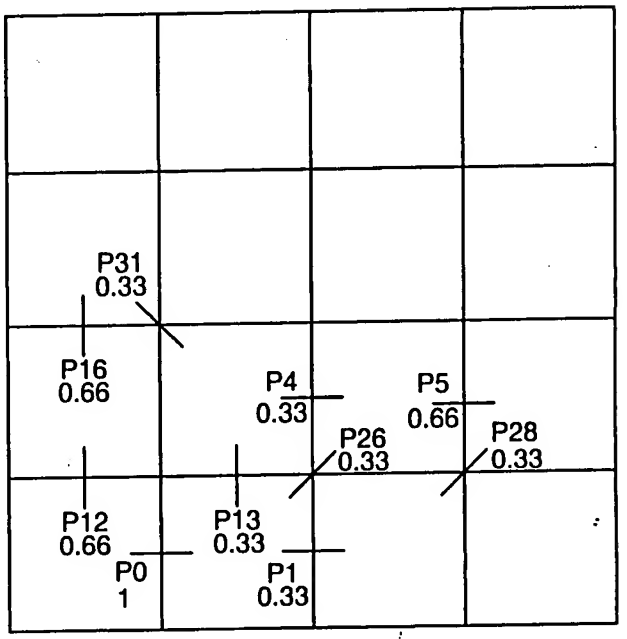

Figure 28

20050124-5950-PCT-00

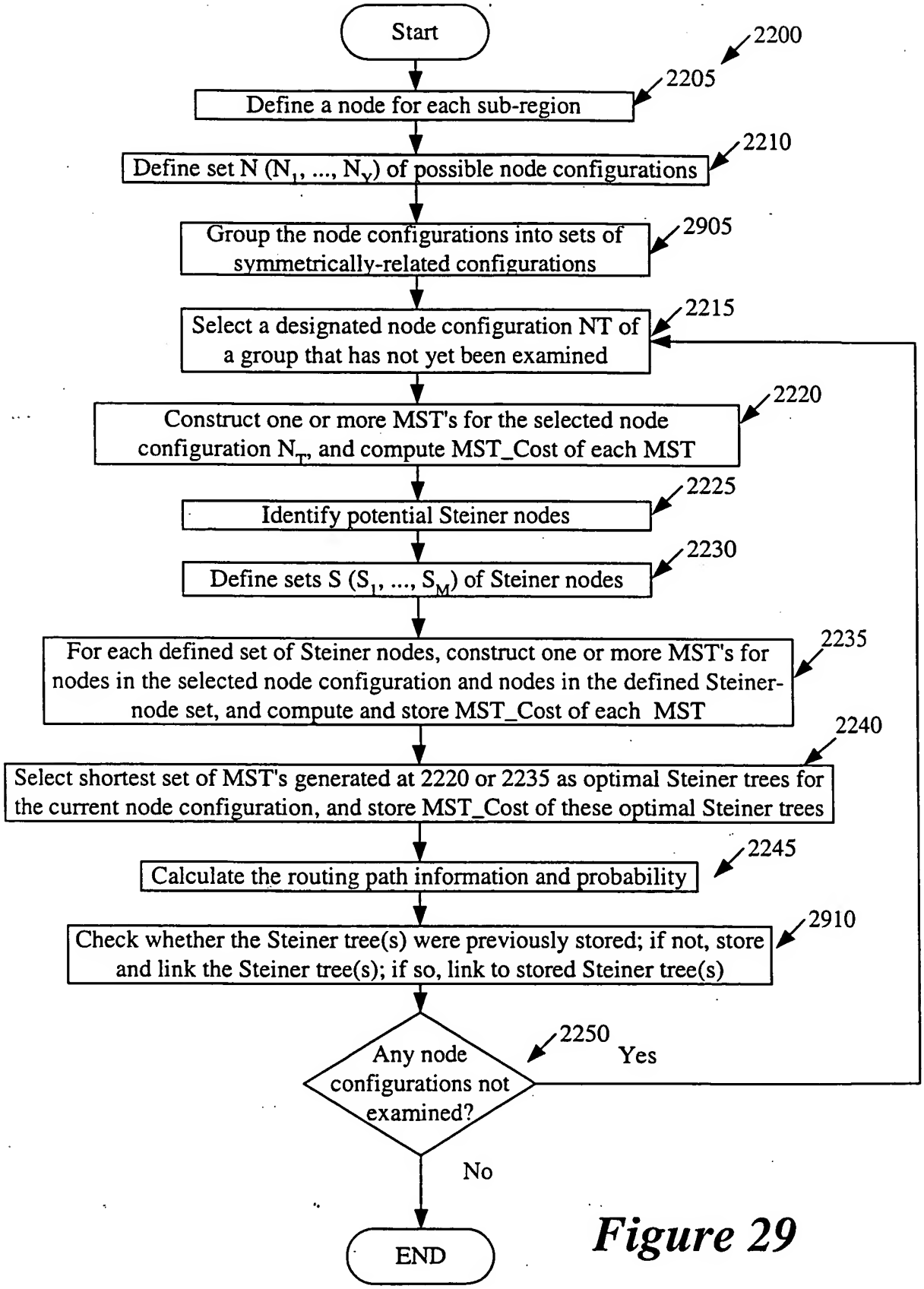

Figure 29

20501O"E960+00T

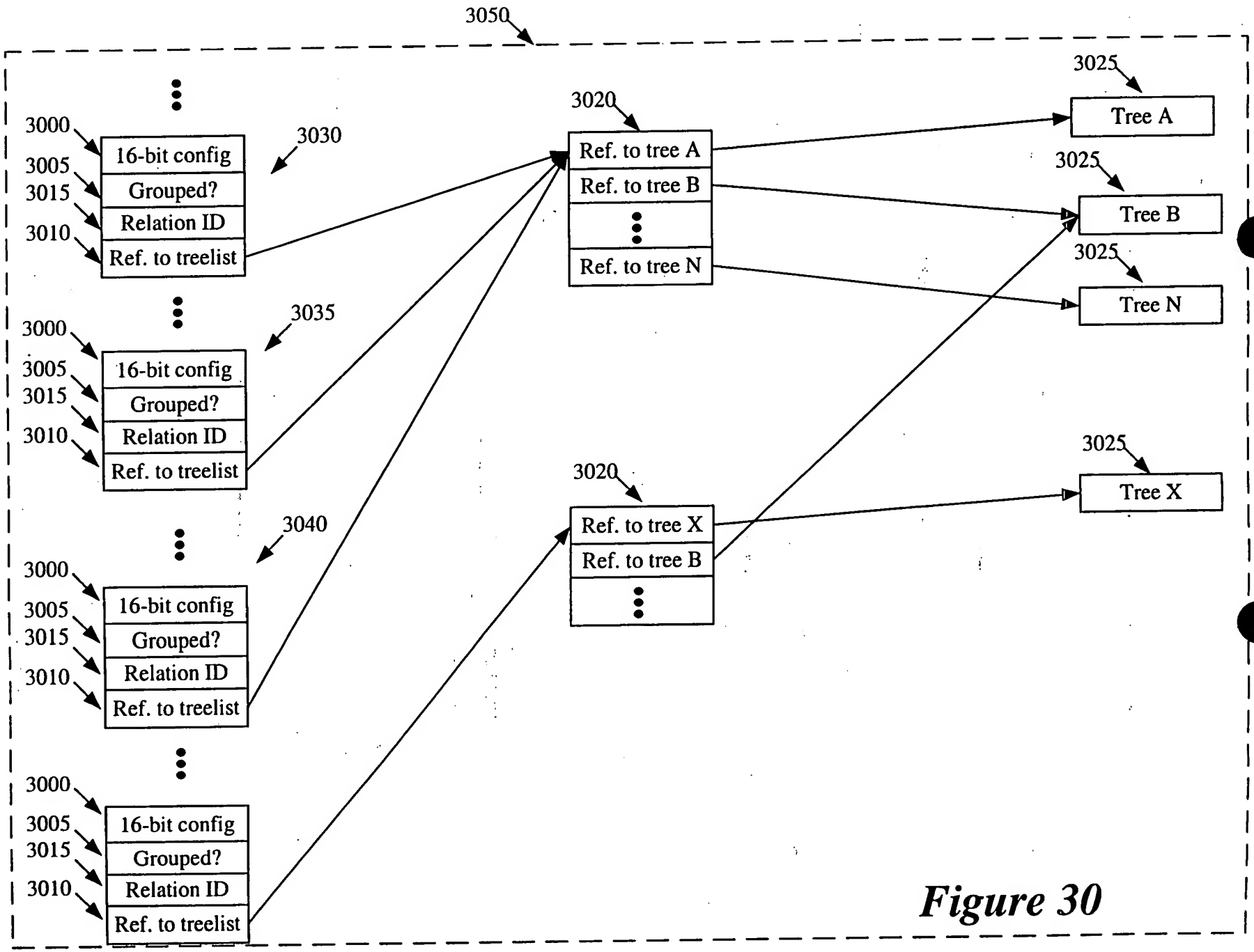

Figure 30

20050154604A1

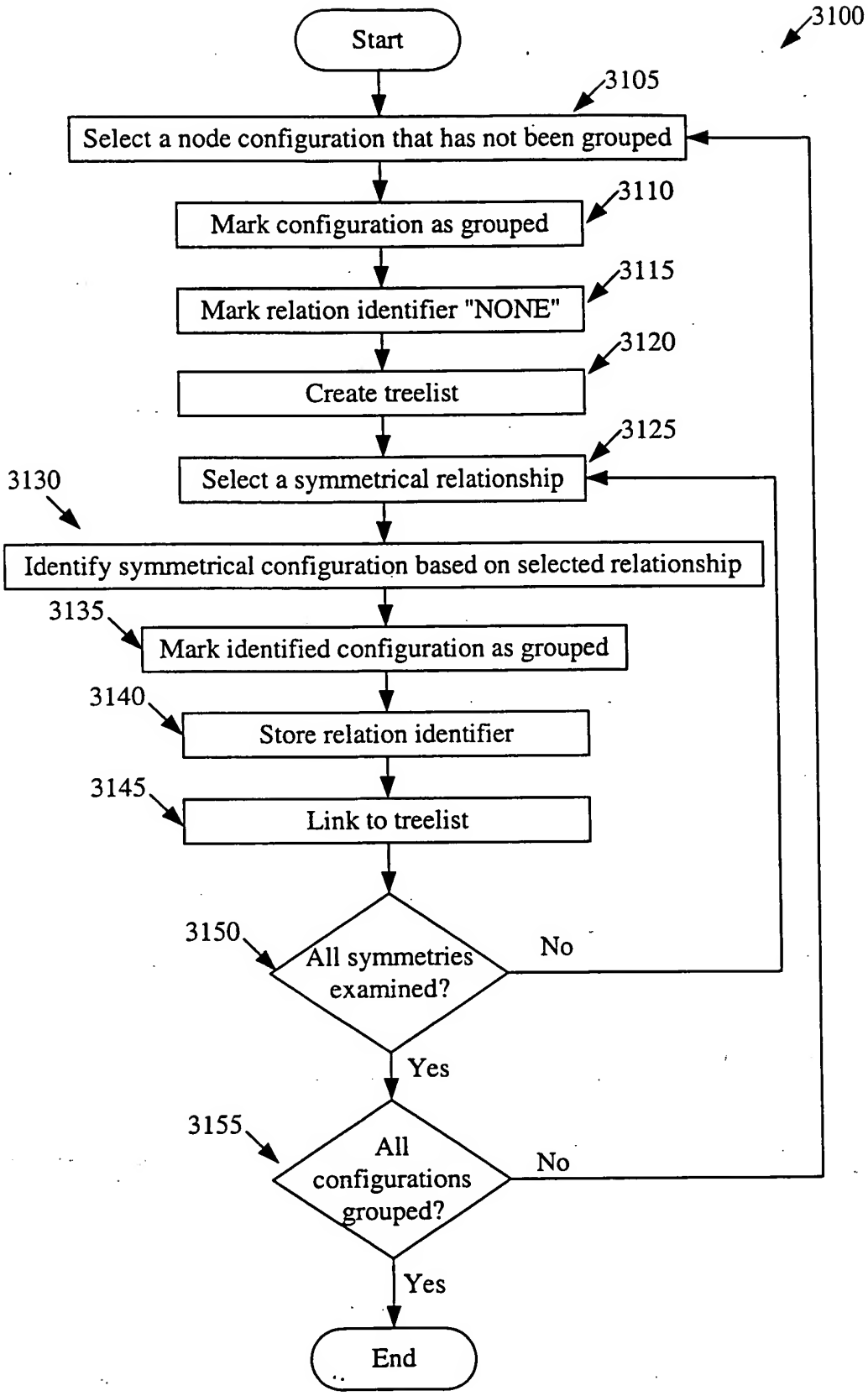

Figure 31

20501D "E960h00T

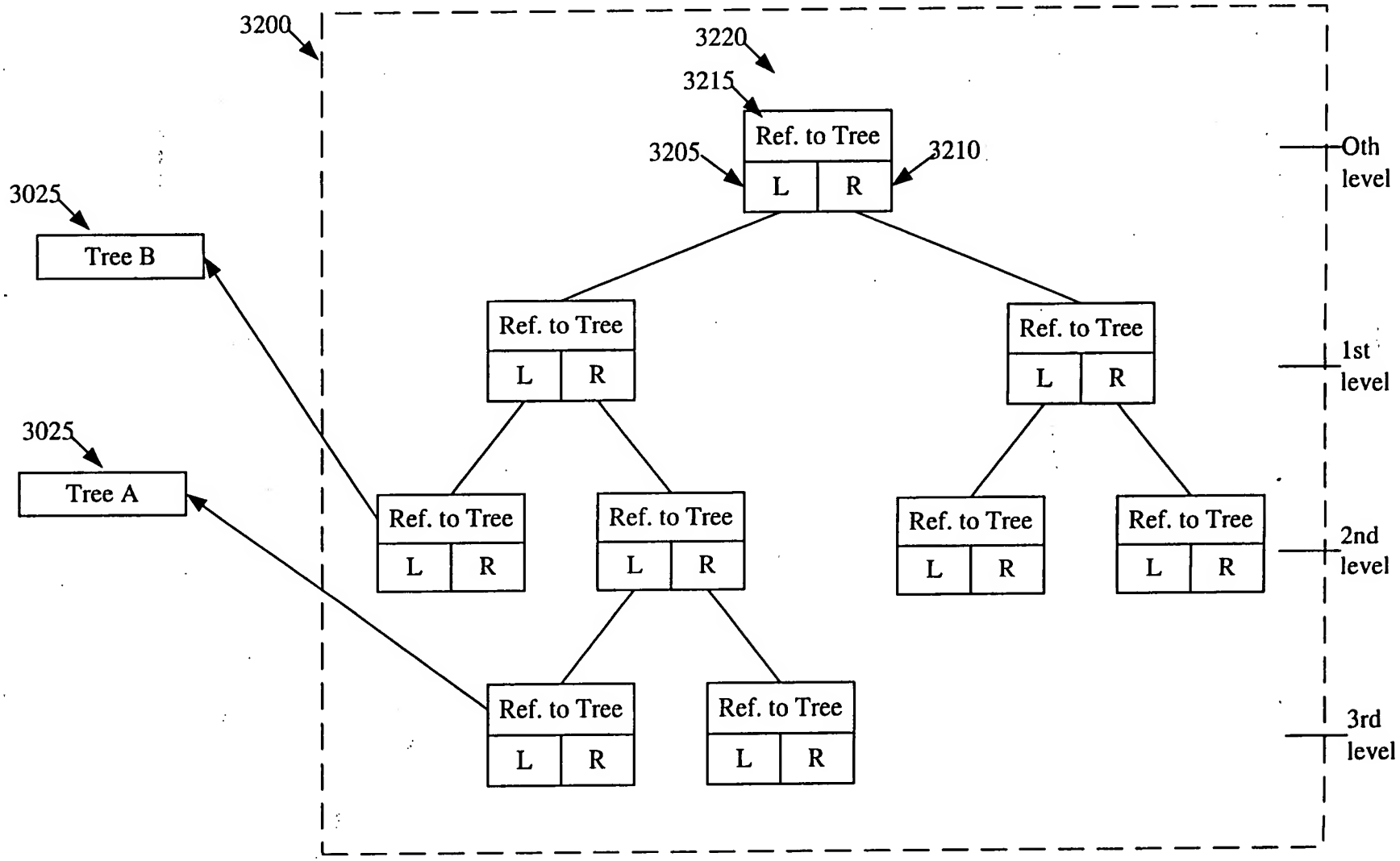

Figure 32

20050154984

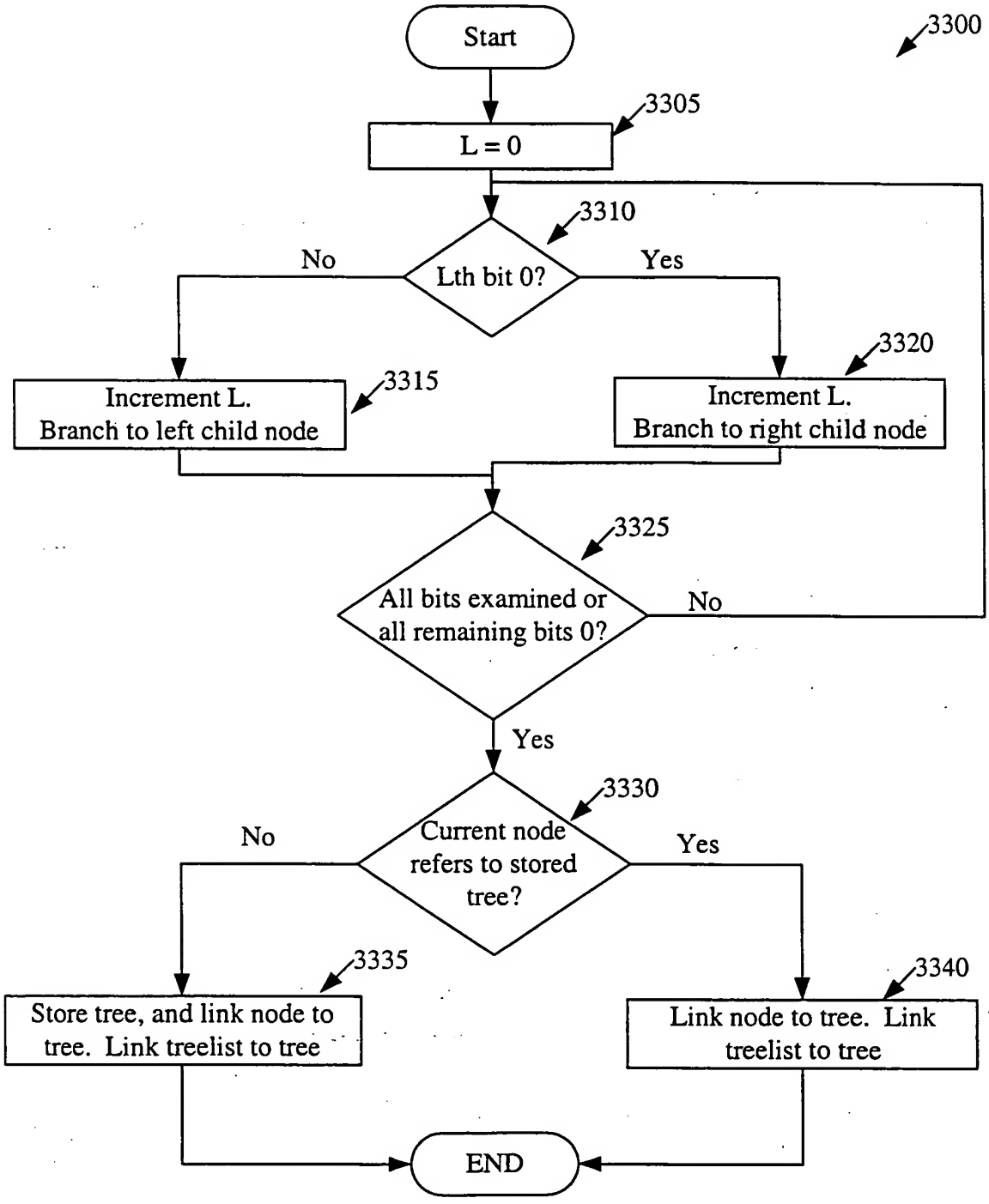

Figure 33

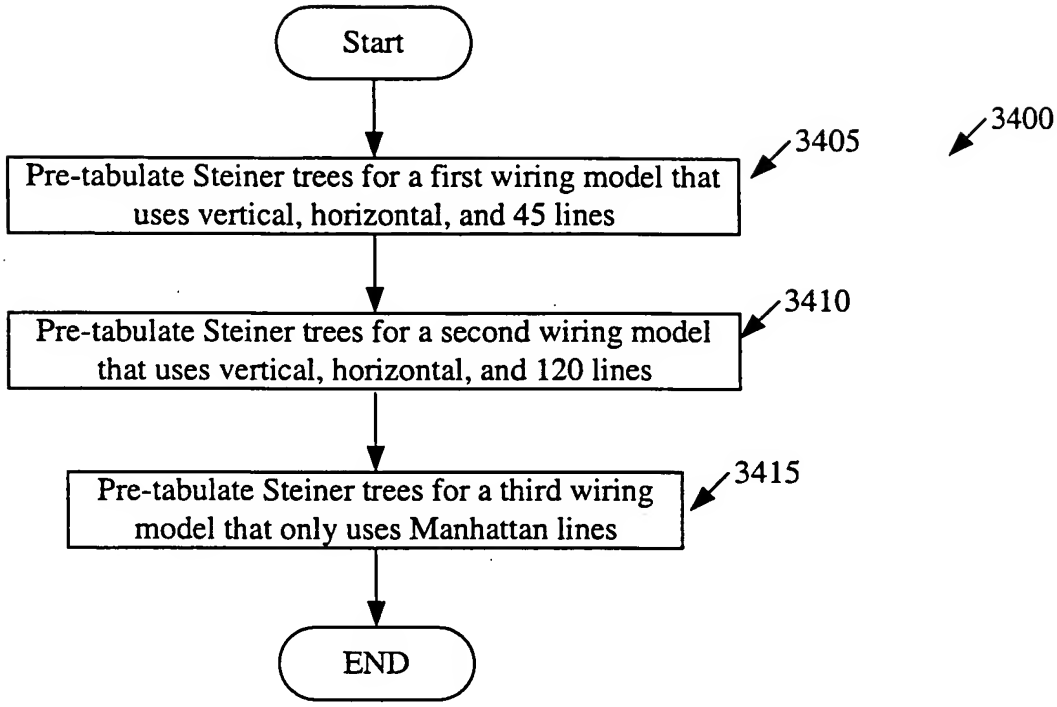

Figure 34

20050734.050502

20501O "E960hO0F

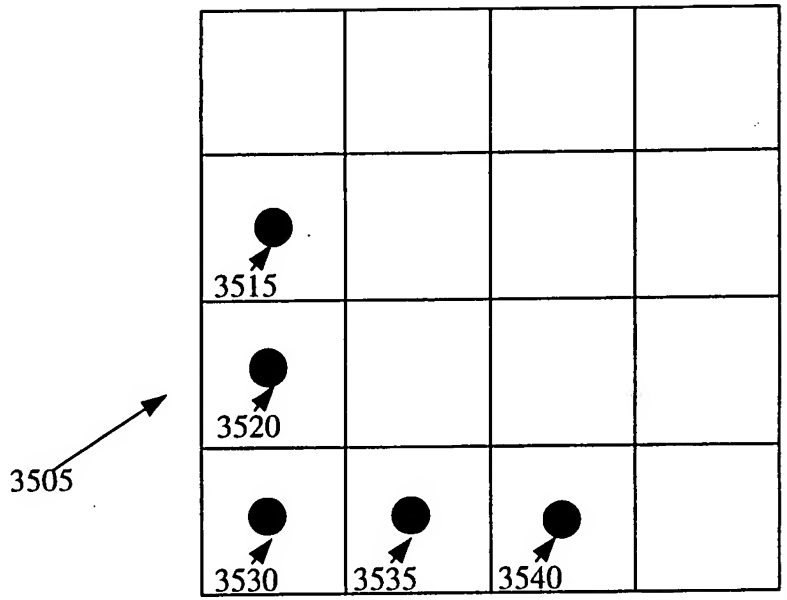

Figure 35A

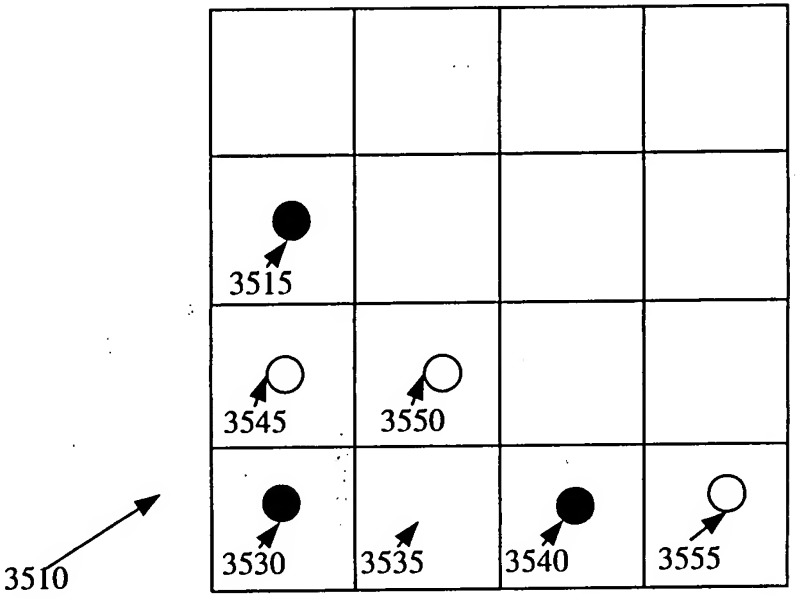

Figure 35B

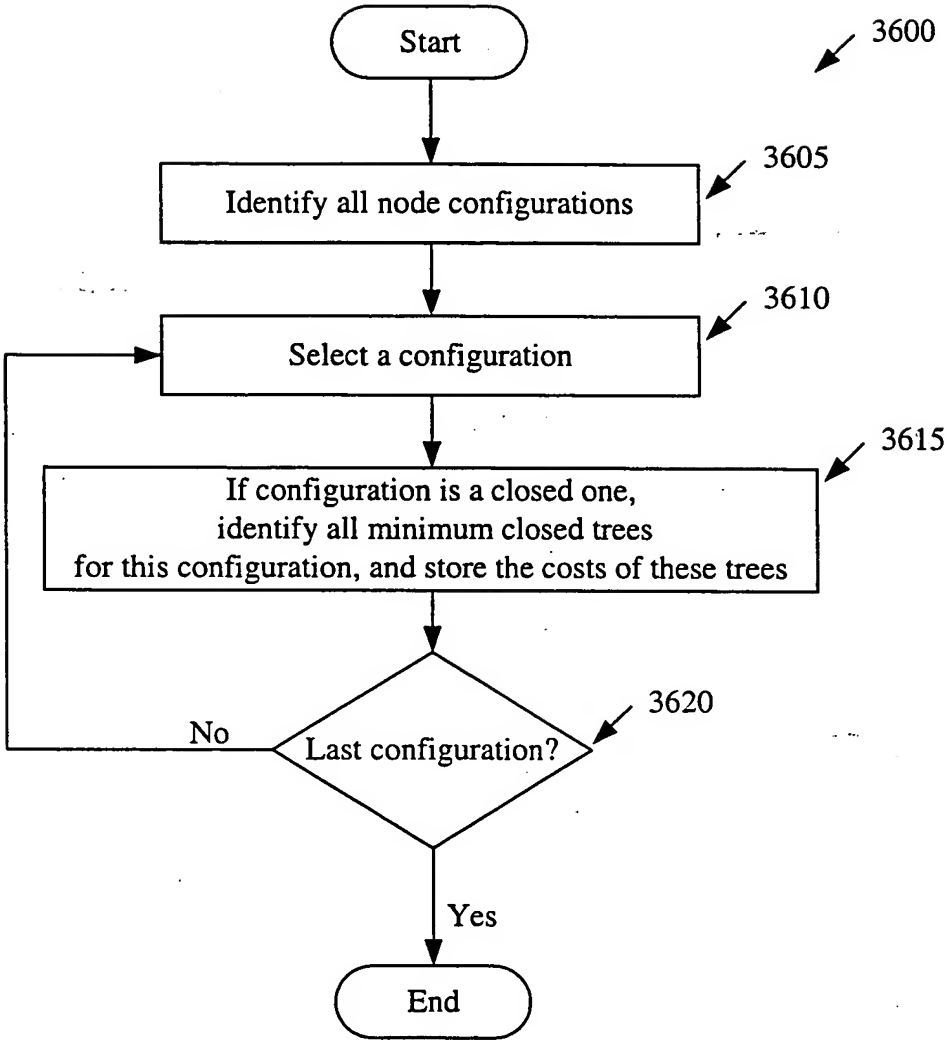
Start
3600
Identify all node configurations
3605
Select a configuration
3610
If configuration is a closed one,
identify all minimum closed trees
for this configuration, and store the costs of these trees
3615
Last configuration?
3620
No
Yes
End

Figure 36

20050154594A1

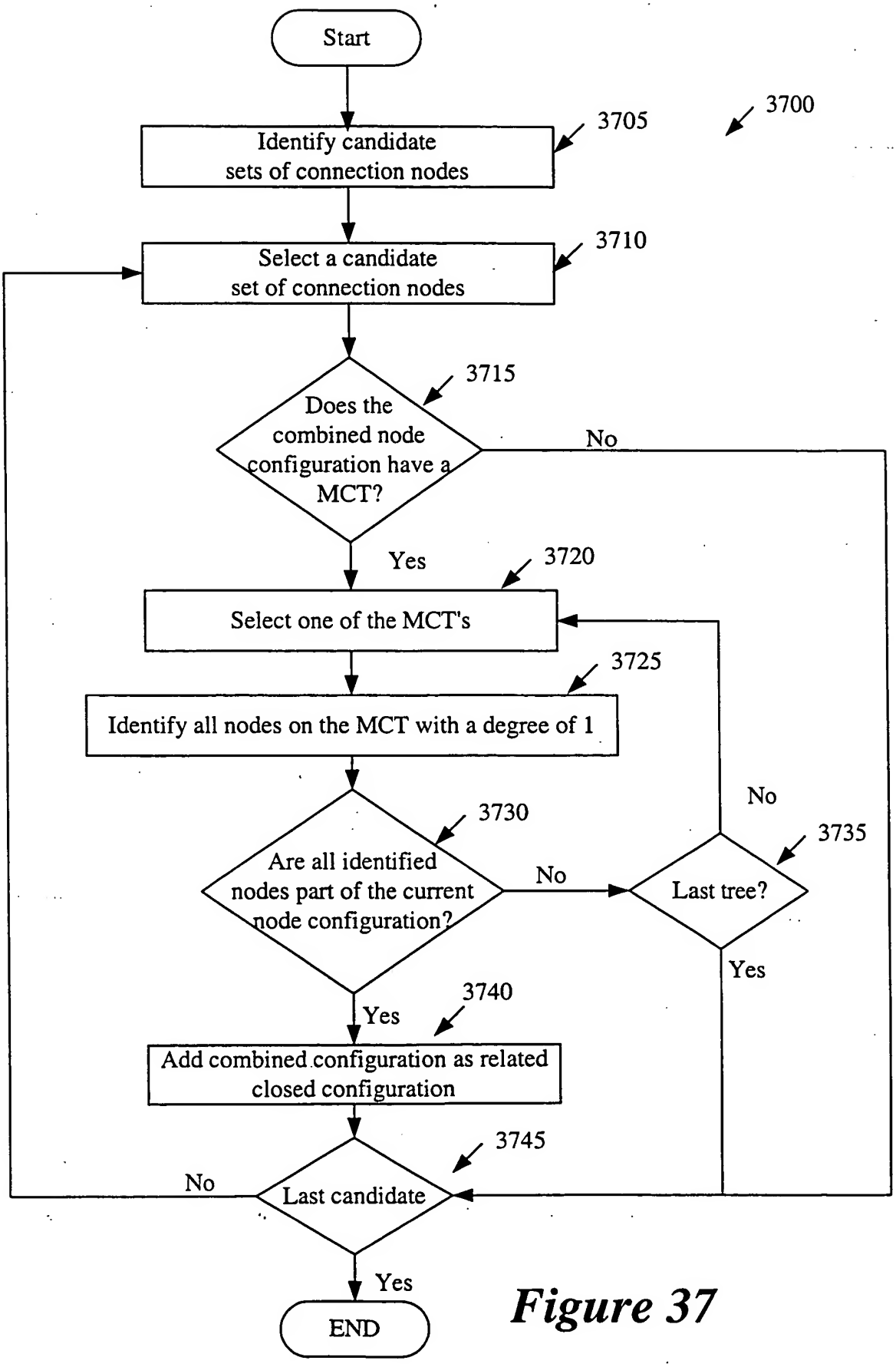

Figure 37

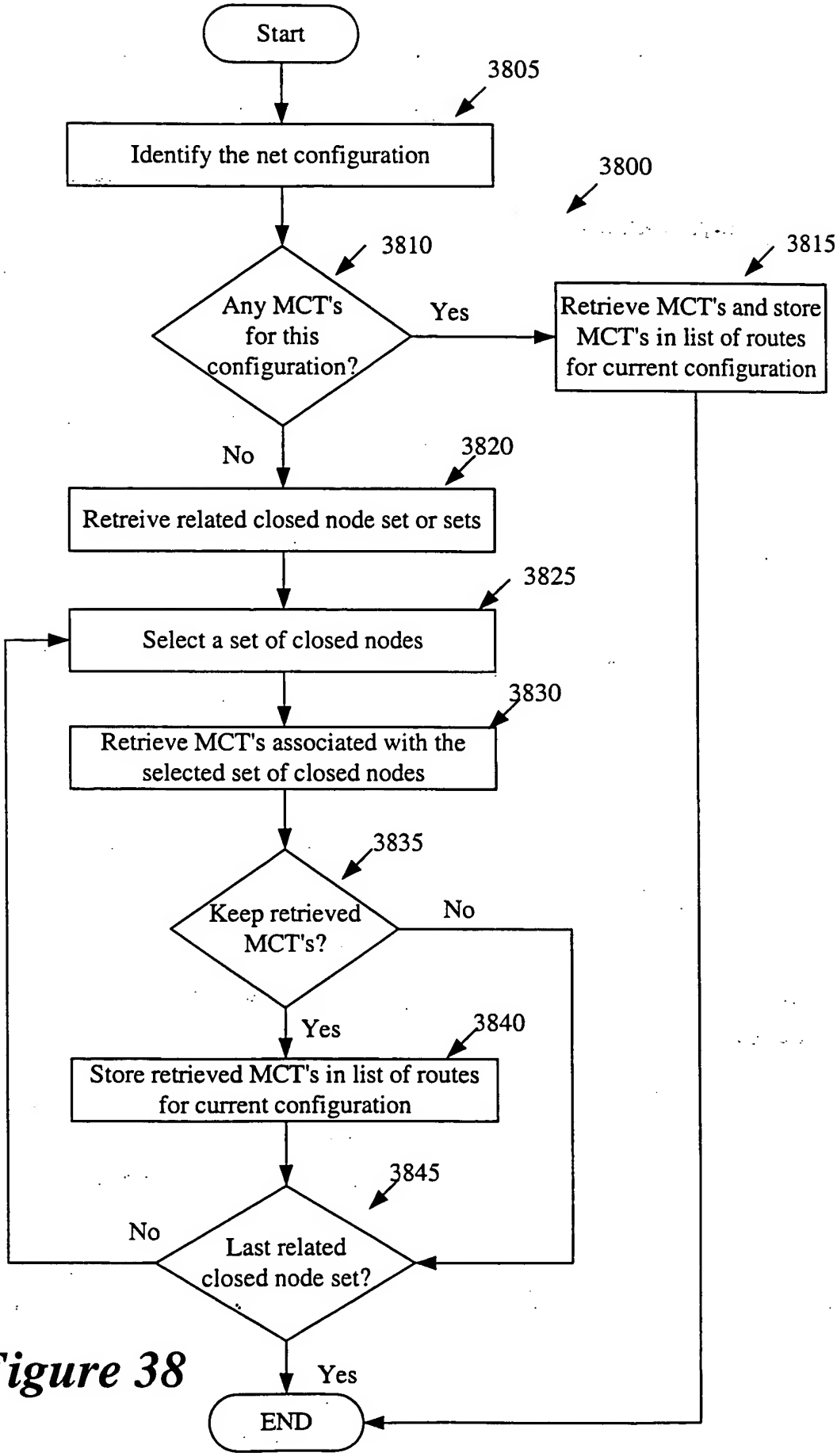
Start
3805
Identify the net configuration
3800
3810
Any MCT's for this configuration?
Yes
3815
Retrieve MCT's and store MCT's in list of routes for current configuration
No
3820
Retreive related closed node set or sets
3825
Select a set of closed nodes
3830
Retrieve MCT's associated with the selected set of closed nodes
3835
Keep retrieved MCT's?
No
Yes
3840
Store retrieved MCT's in list of routes for current configuration
3845
No
Last related closed node set?
Yes
END

Figure 38

20501O" E960hOOT

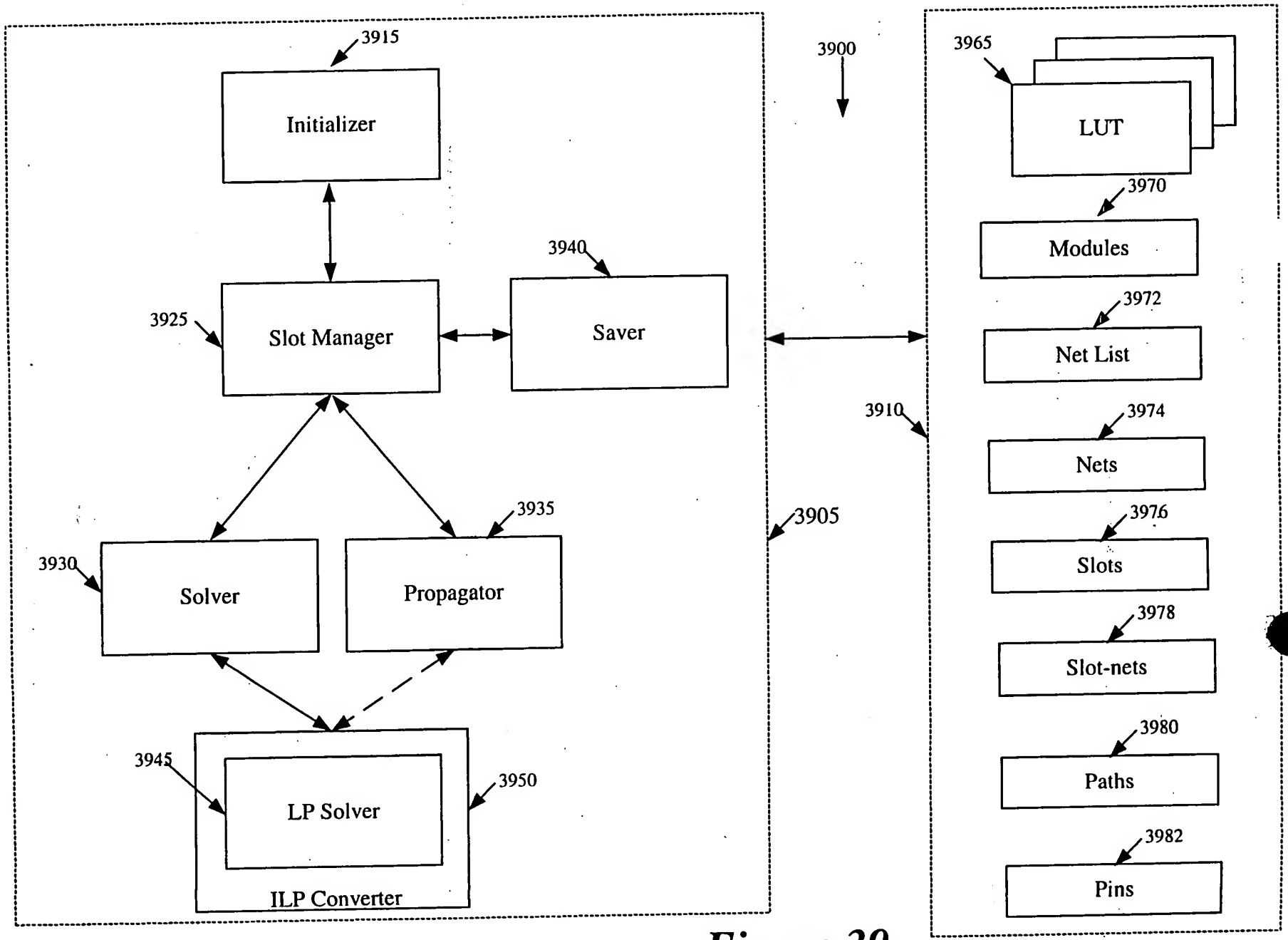

Figure 39

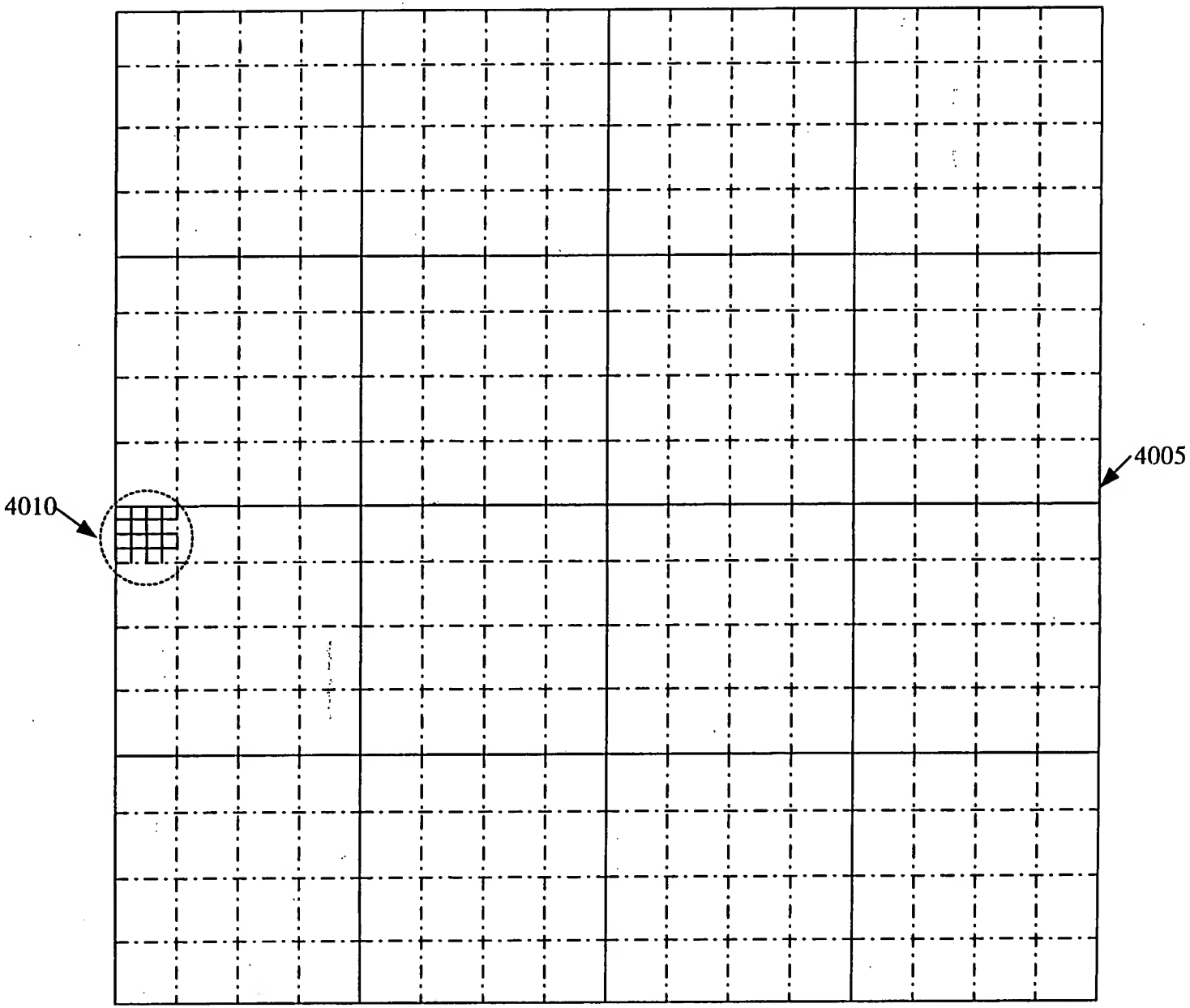

Figure 40

20501O "E960h00T

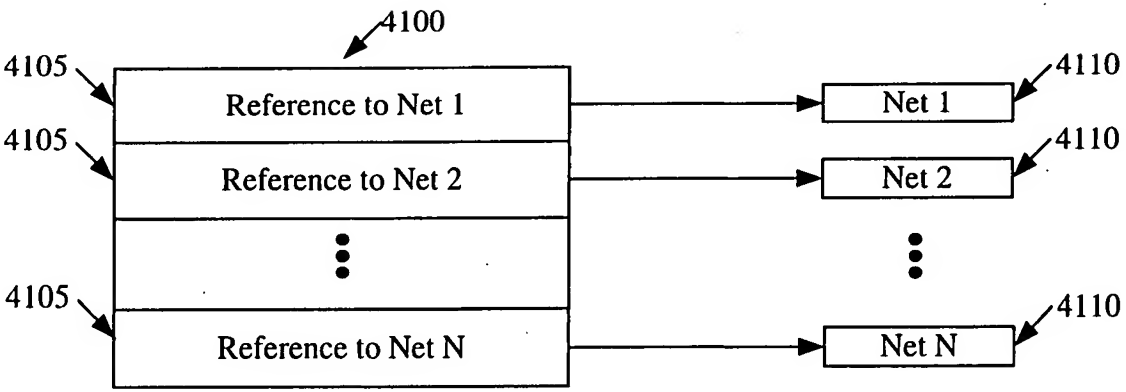

Figure 41

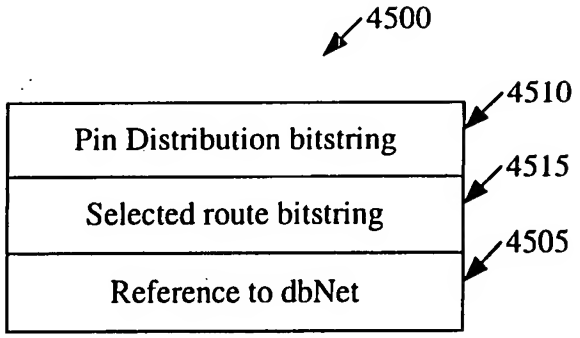

Figure 45

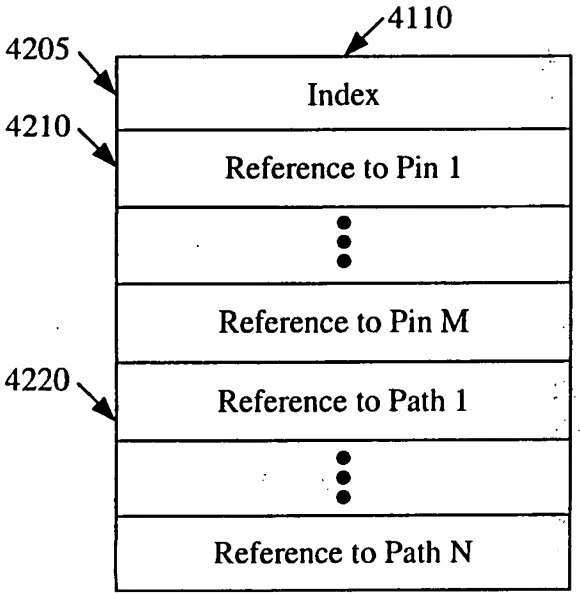

Figure 42

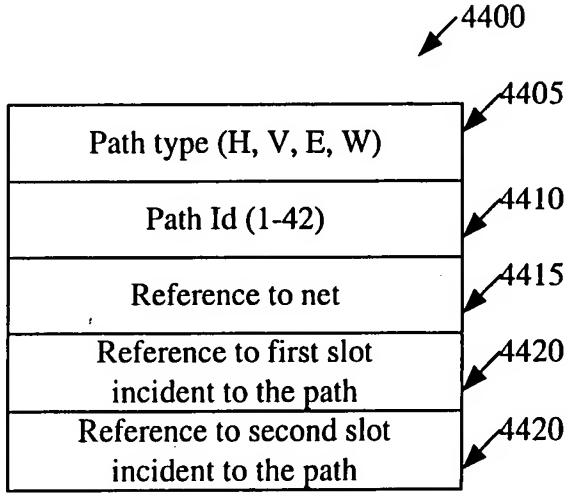

Figure 44

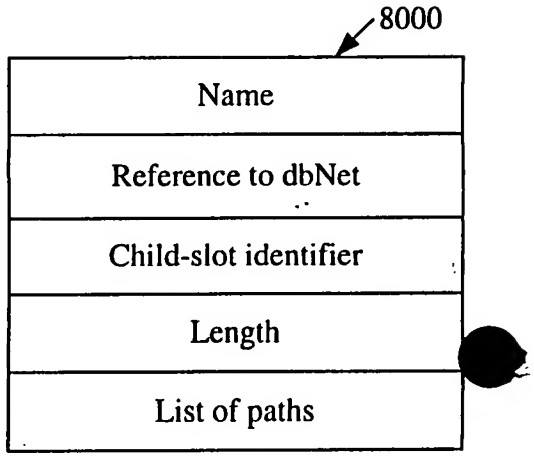

Figure 80

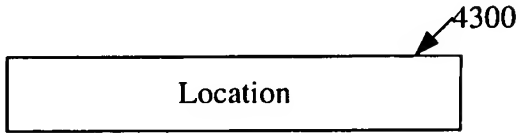

Figure 43

20020"E960h00T

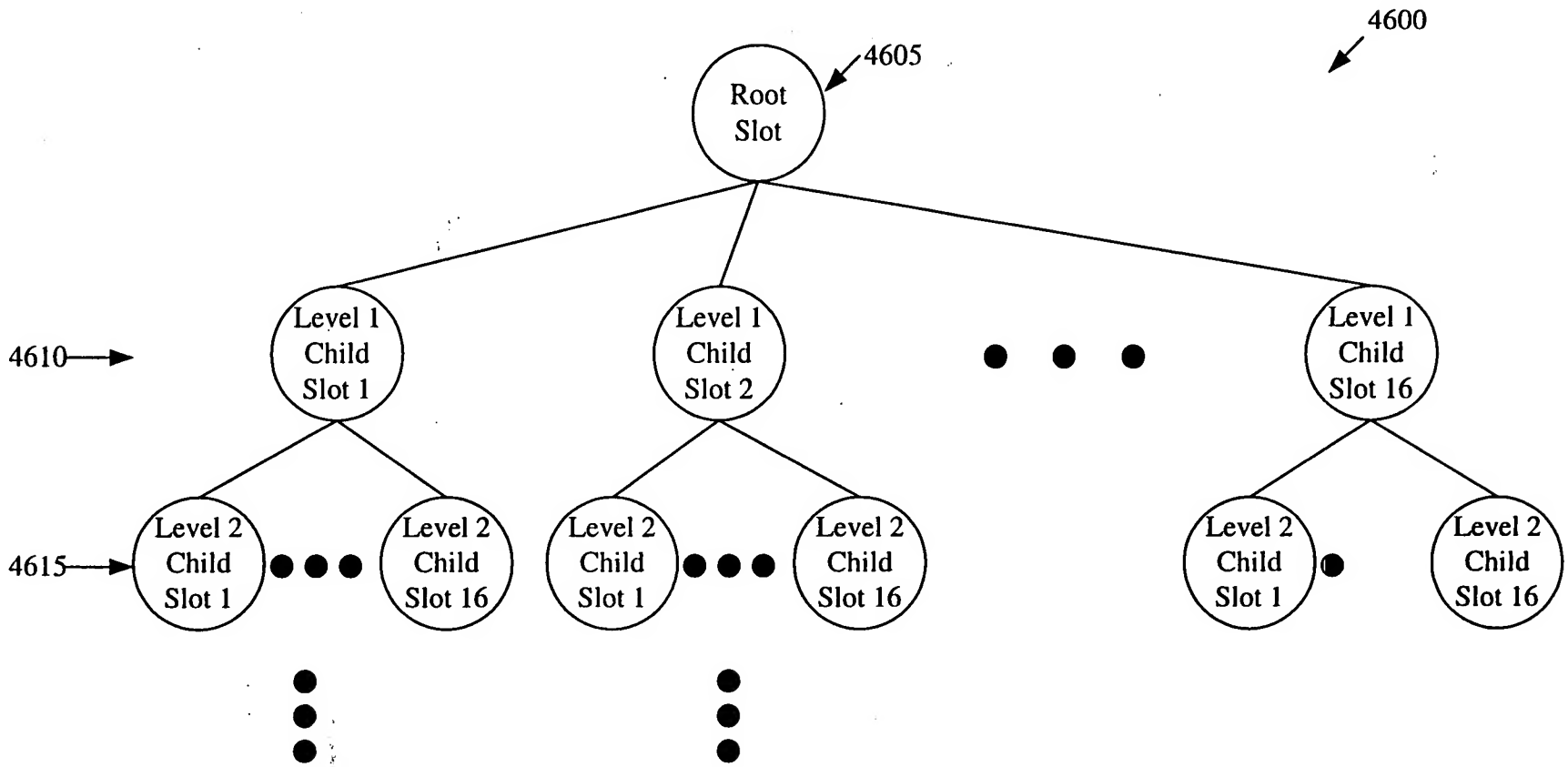

Figure 46

20070"E960h00T

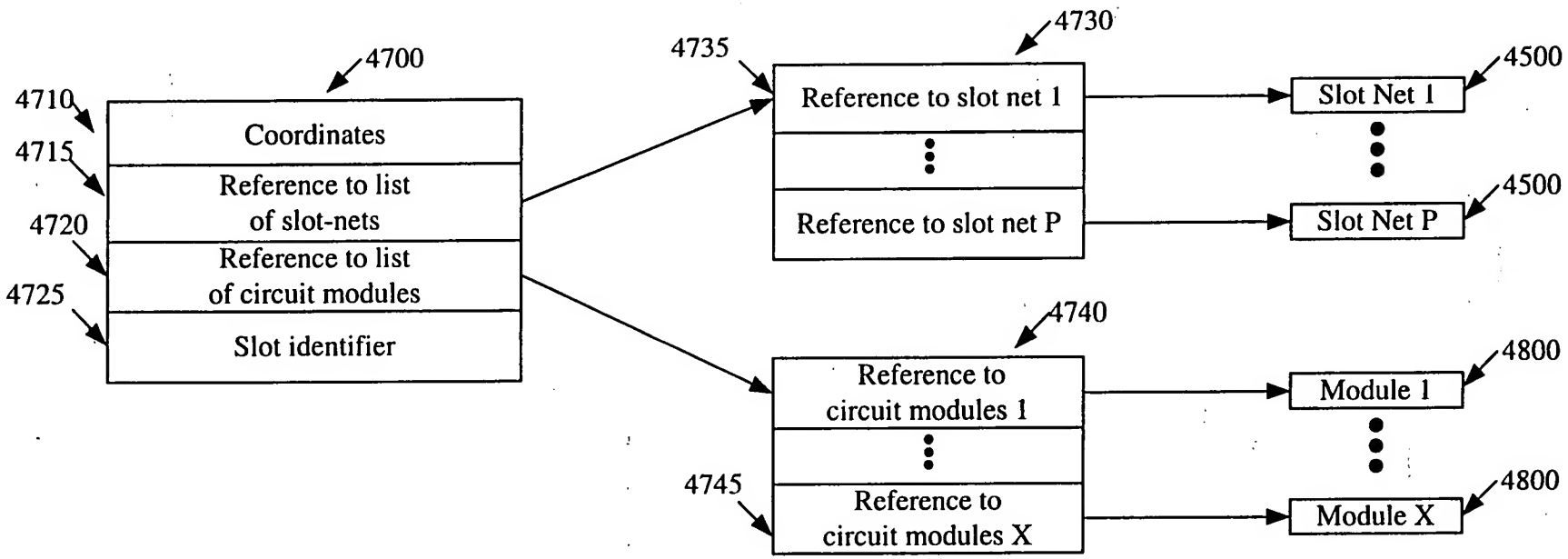

Figure 47

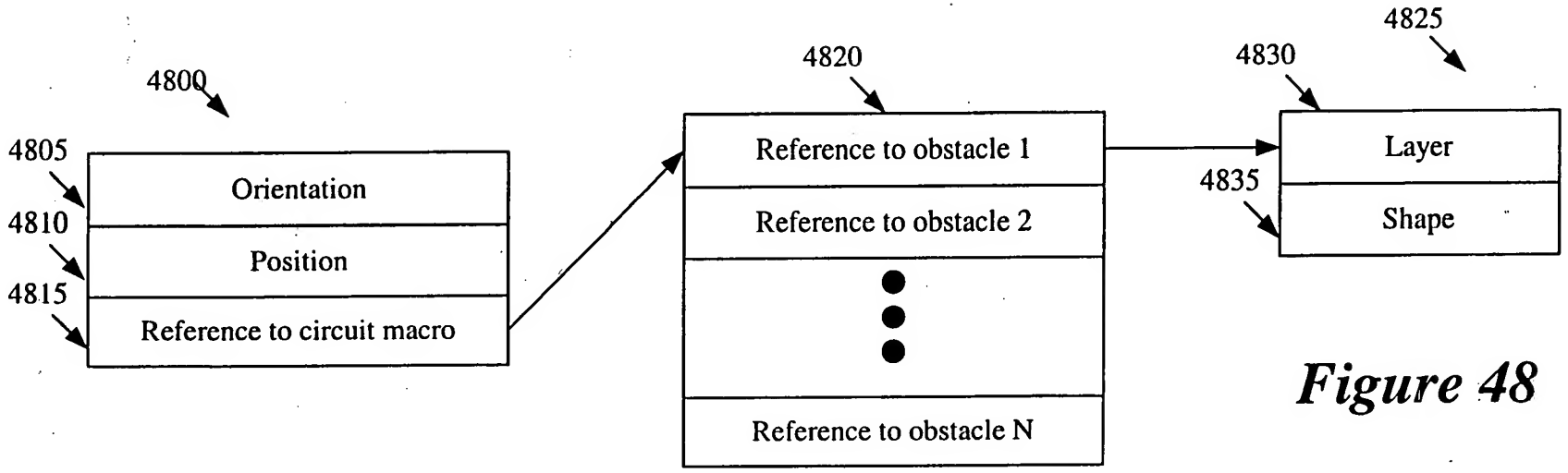

Figure 48

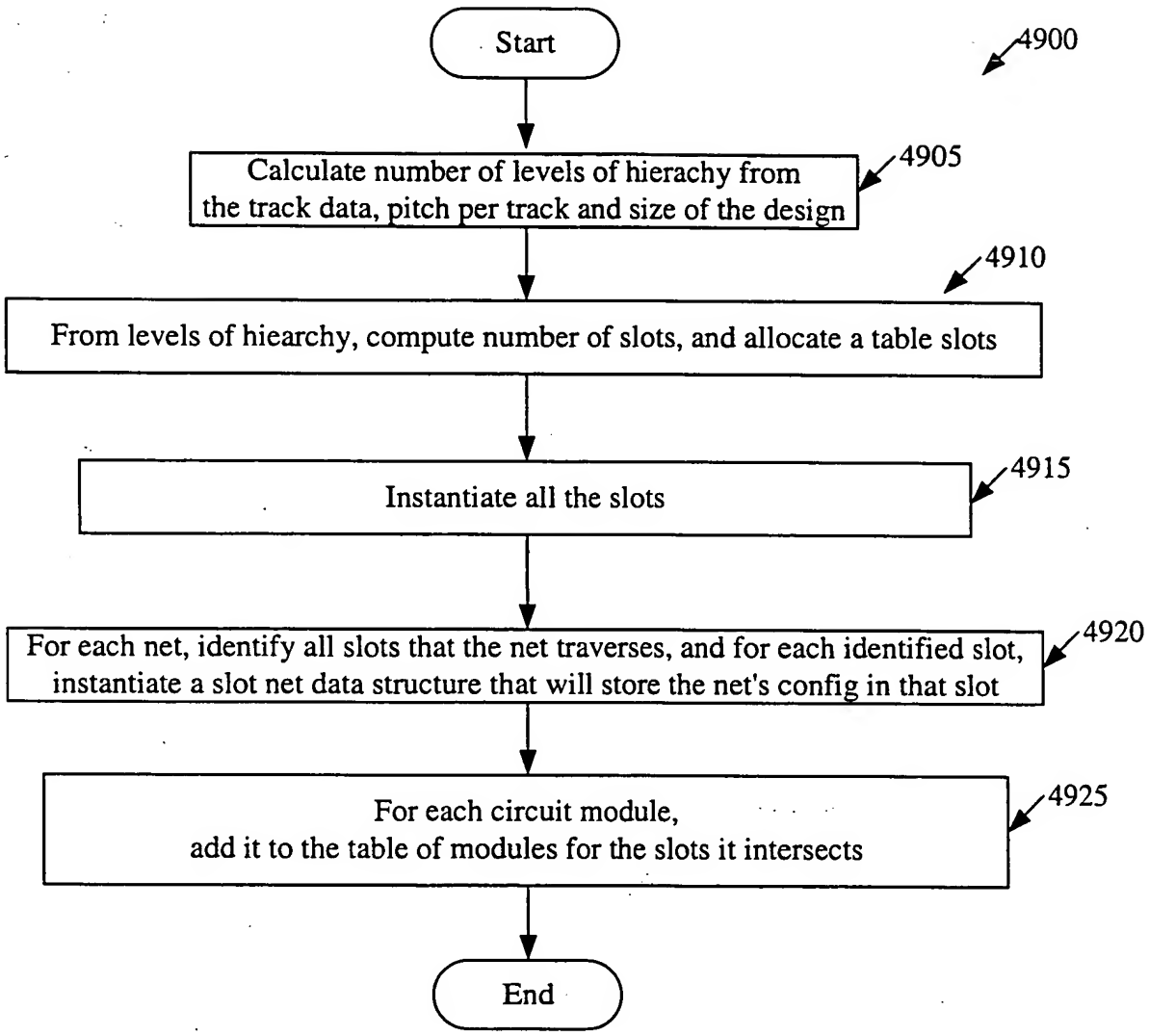
Start
4900
Calculate number of levels of hierachy from the track data, pitch per track and size of the design
4905
4910
From levels of hiearchy, compute number of slots, and allocate a table slots
Instantiate all the slots
4915
For each net, identify all slots that the net traverses, and for each identified slot, instantiate a slot net data structure that will store the net's config in that slot
4920
For each circuit module, add it to the table of modules for the slots it intersects
4925
End

Figure 49

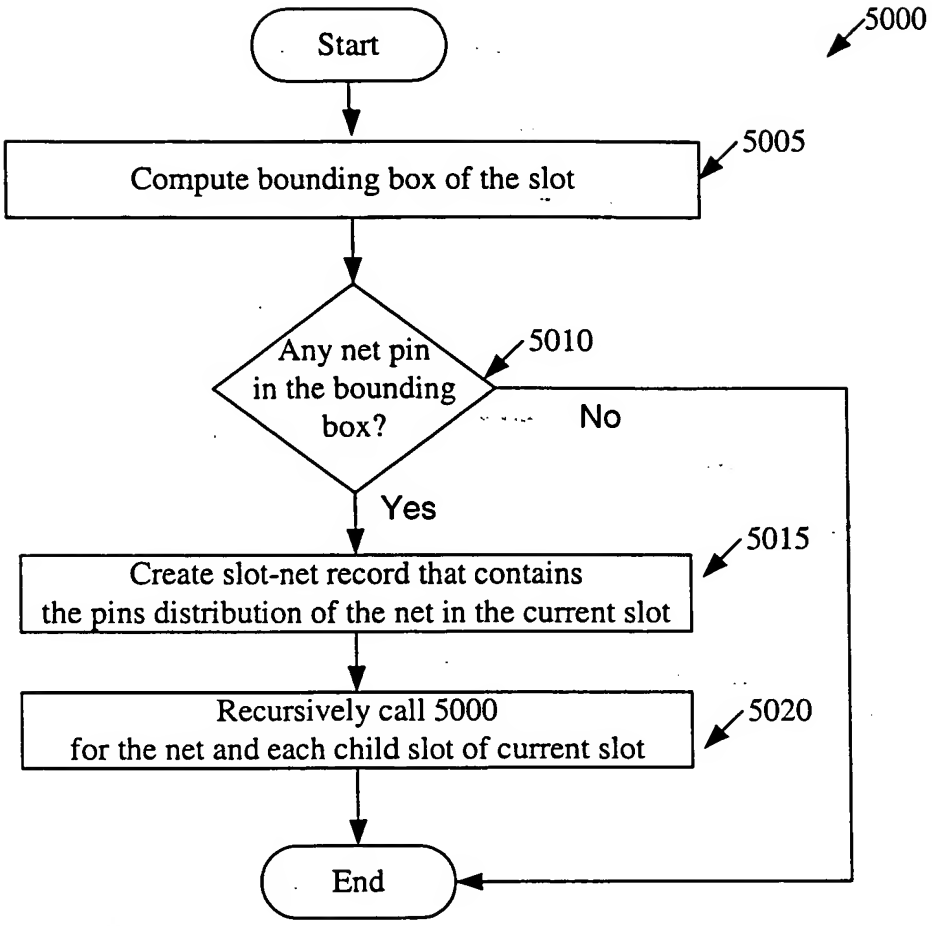
5000
Start
Compute bounding box of the slot
5005
Any net pin in the bounding box?
5010
No
Yes
Create slot-net record that contains the pins distribution of the net in the current slot
5015
Recursively call 5000 for the net and each child slot of current slot
5020
End

Figure 50

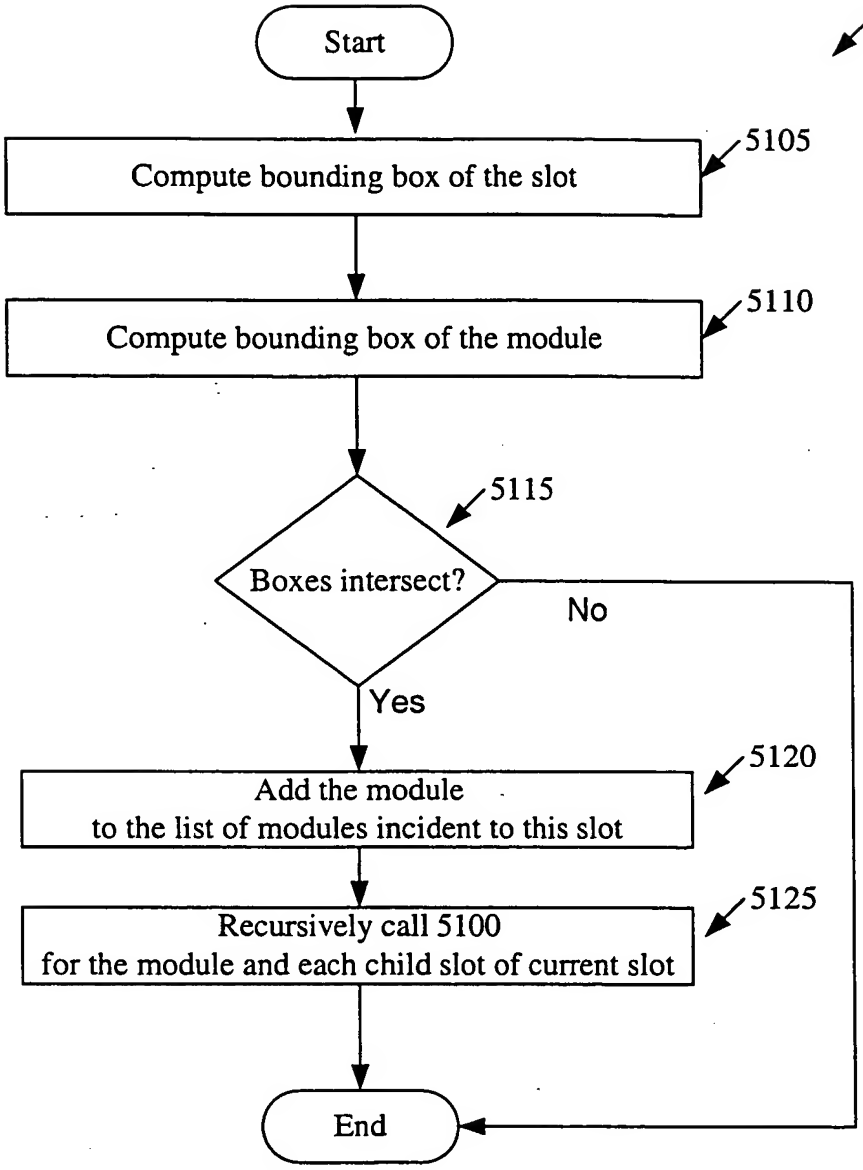
Start
5100
Compute bounding box of the slot
5105
Compute bounding box of the module
5110
5115
Boxes intersect?
No
Yes
Add the module
to the list of modules incident to this slot
5120
Recursively call 5100
for the module and each child slot of current slot
5125
End

Figure 51

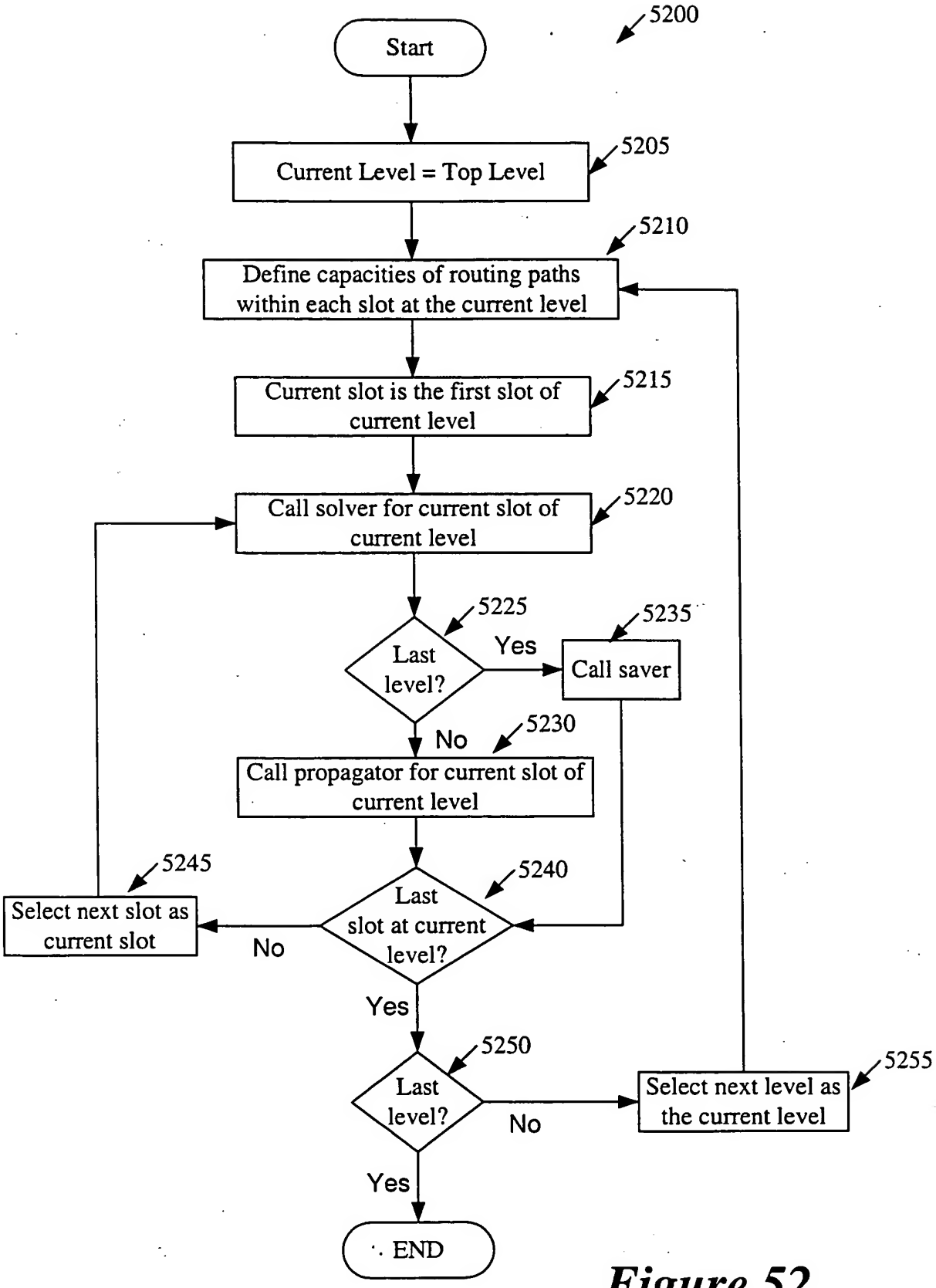
5200
Start
Current Level = Top Level
5205
Define capacities of routing paths within each slot at the current level
5210
Current slot is the first slot of current level
5215
Call solver for current slot of current level
5220
Last level?
5225
Yes
Call saver
5235
No
Call propagator for current slot of current level
5230
Last slot at current level?
5240
No
Select next slot as current slot
5245
Yes
Last level?
5250
No
Select next level as the current level
5255
Yes
END

Figure 52

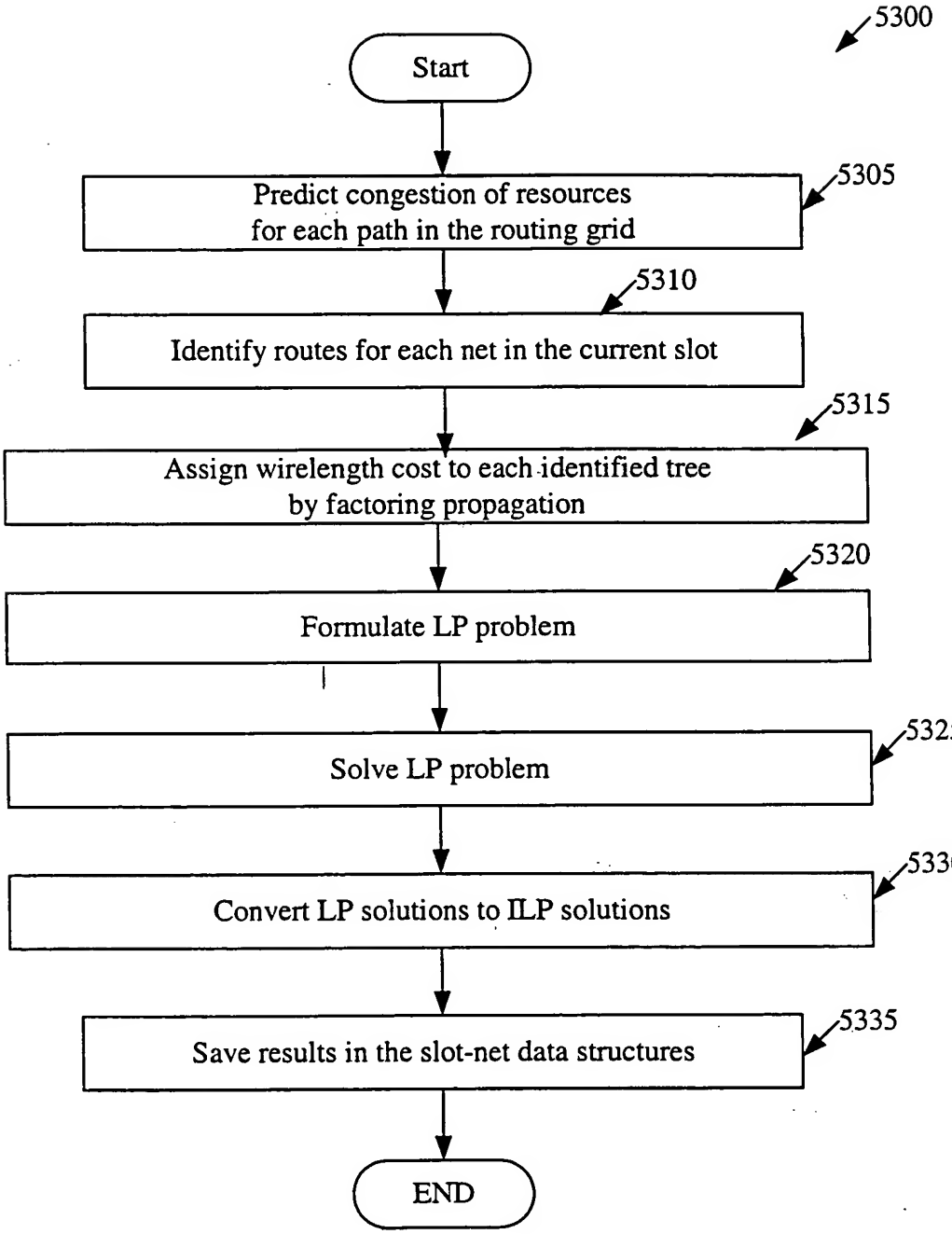
5300
Start
Predict congestion of resources for each path in the routing grid
5305
5310
Identify routes for each net in the current slot
5315
Assign wirelength cost to each identified tree by factoring propagation
5320
Formulate LP problem
5325
Solve LP problem
5330
Convert LP solutions to ILP solutions
5335
Save results in the slot-net data structures
END

Figure 53

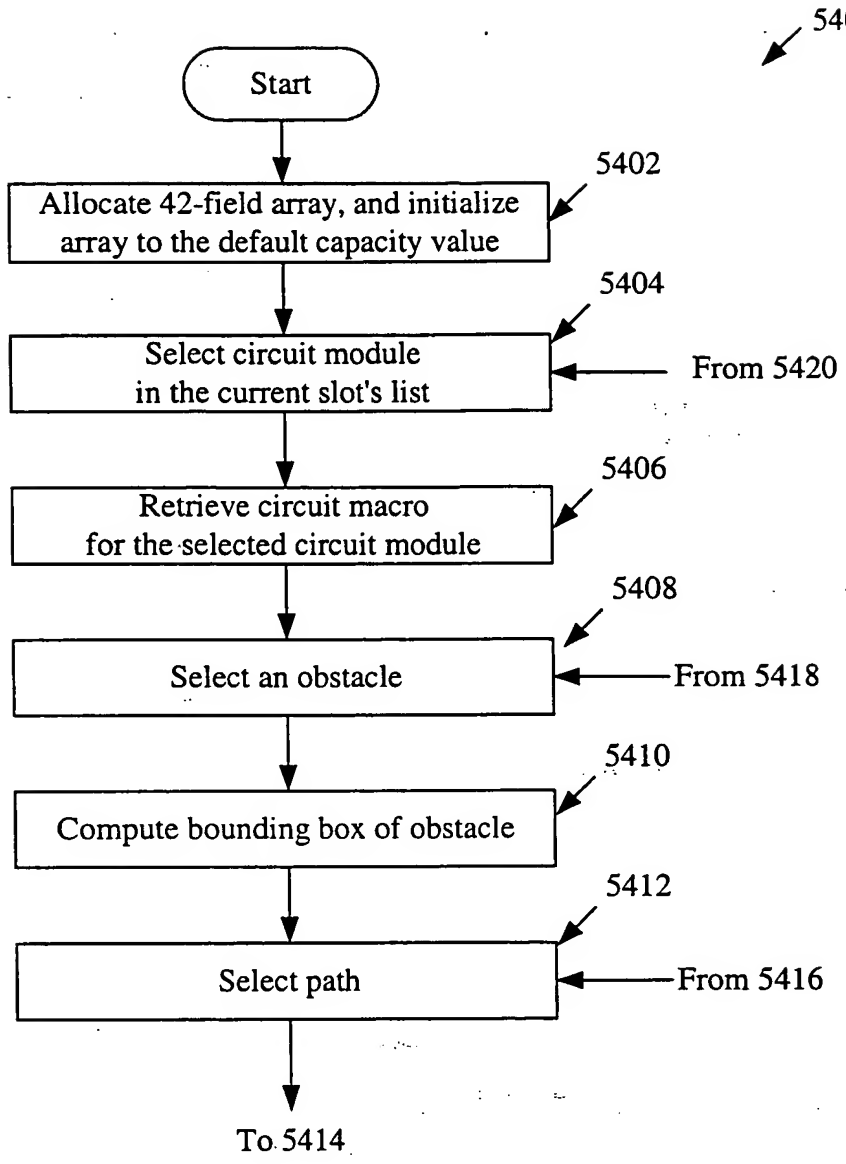

Figure 54A

Figure 54: ***Figure 54A*** / ***Figure 54B***

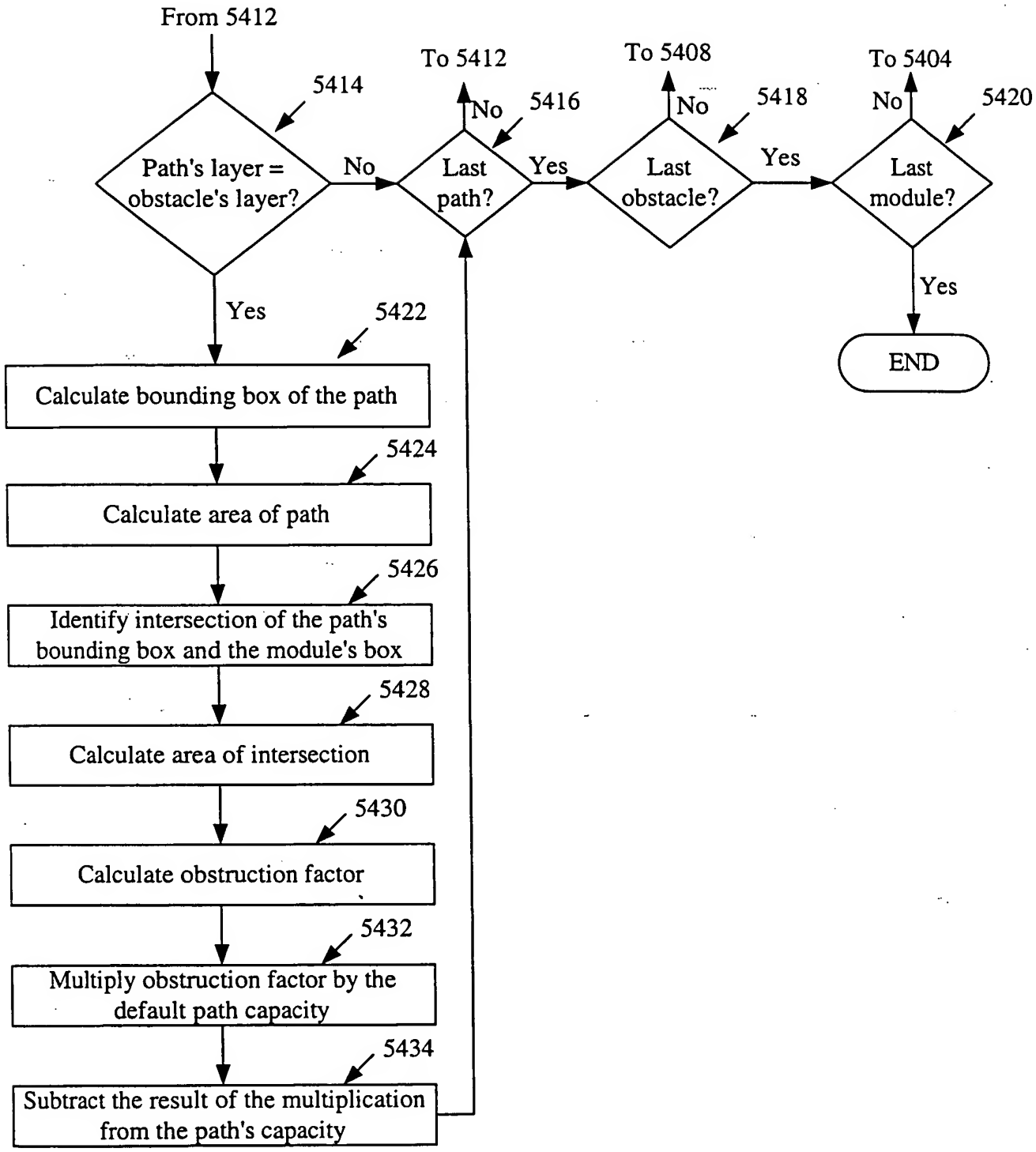
From 5412
5414
Path's layer = obstacle's layer?
No
To 5412
No
5416
Last path?
Yes
To 5408
No
5418
Last obstacle?
Yes
To 5404
No
5420
Last module?
Yes
END
Yes
5422
Calculate bounding box of the path
5424
Calculate area of path
5426
Identify intersection of the path's bounding box and the module's box
5428
Calculate area of intersection
5430
Calculate obstruction factor
5432
Multiply obstruction factor by the default path capacity
5434
Subtract the result of the multiplication from the path's capacity

Figure 54B

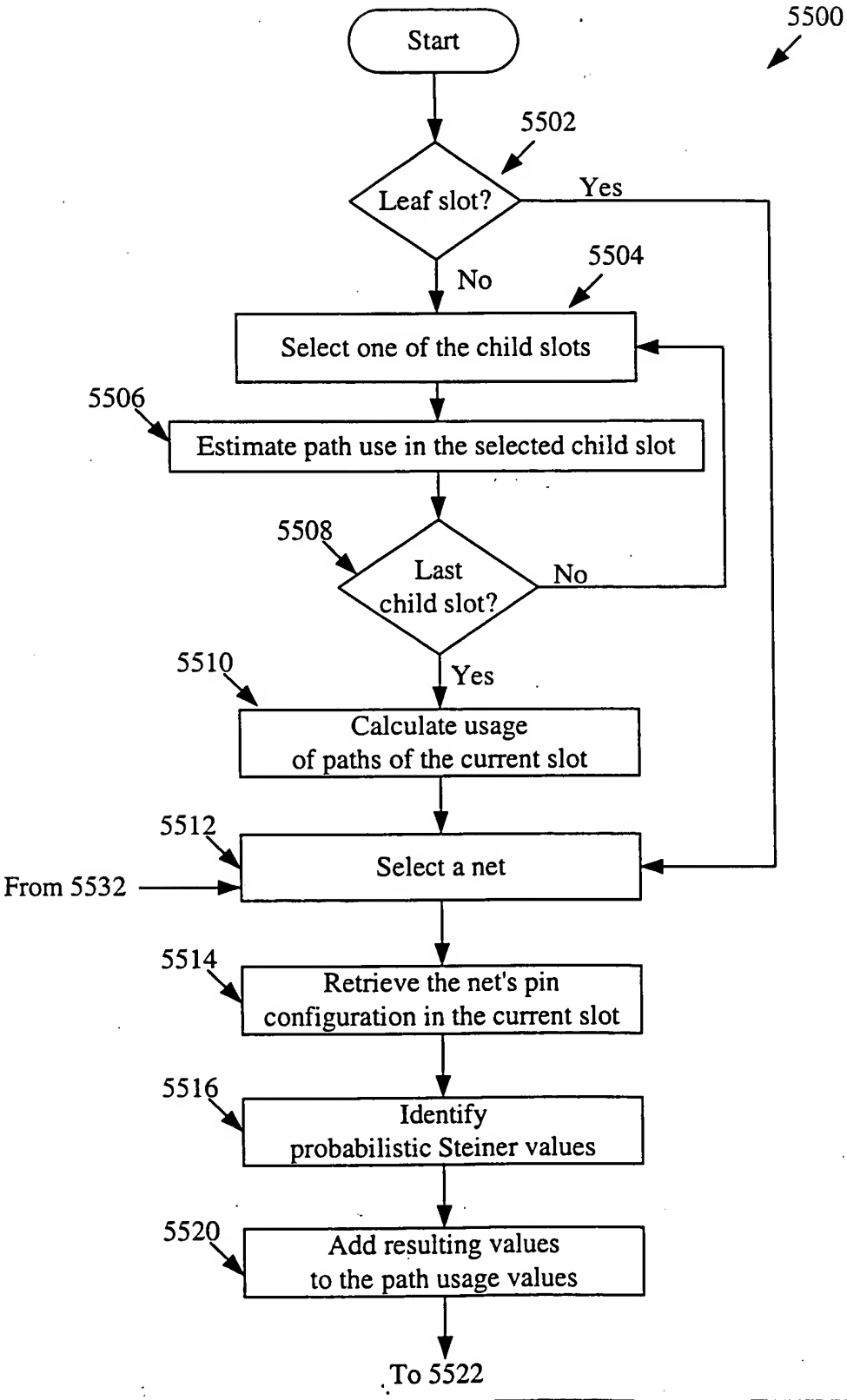

Figure 55A

Figure 55: Figure 55A / Figure 55B

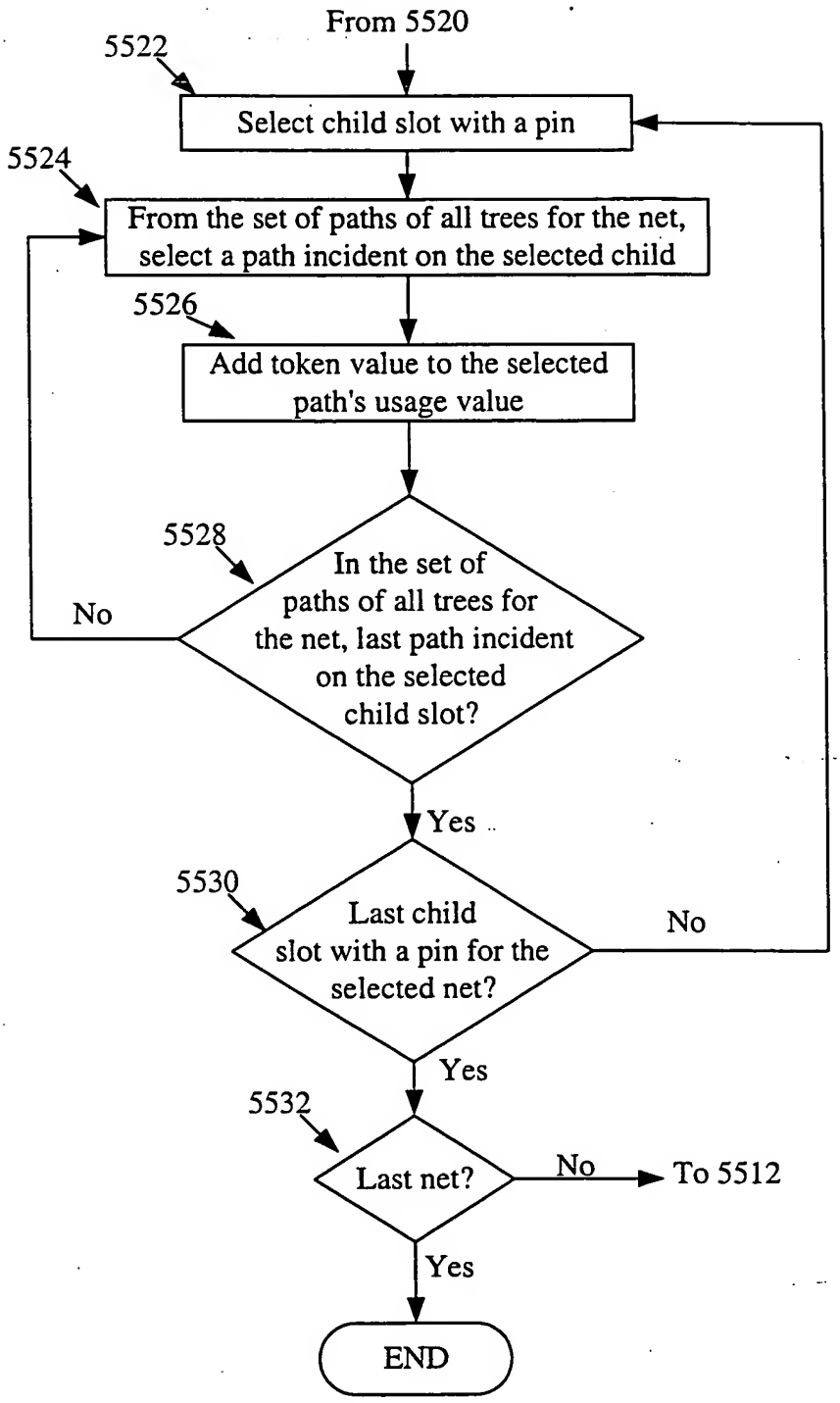
From 5520
5522
Select child slot with a pin
5524
From the set of paths of all trees for the net, select a path incident on the selected child
5526
Add token value to the selected path's usage value
5528
In the set of paths of all trees for the net, last path incident on the selected child slot?
No
Yes
5530
Last child slot with a pin for the selected net?
No
Yes
5532
Last net?
No
To 5512
Yes
END

Figure 55B

20050502.050403.E3696

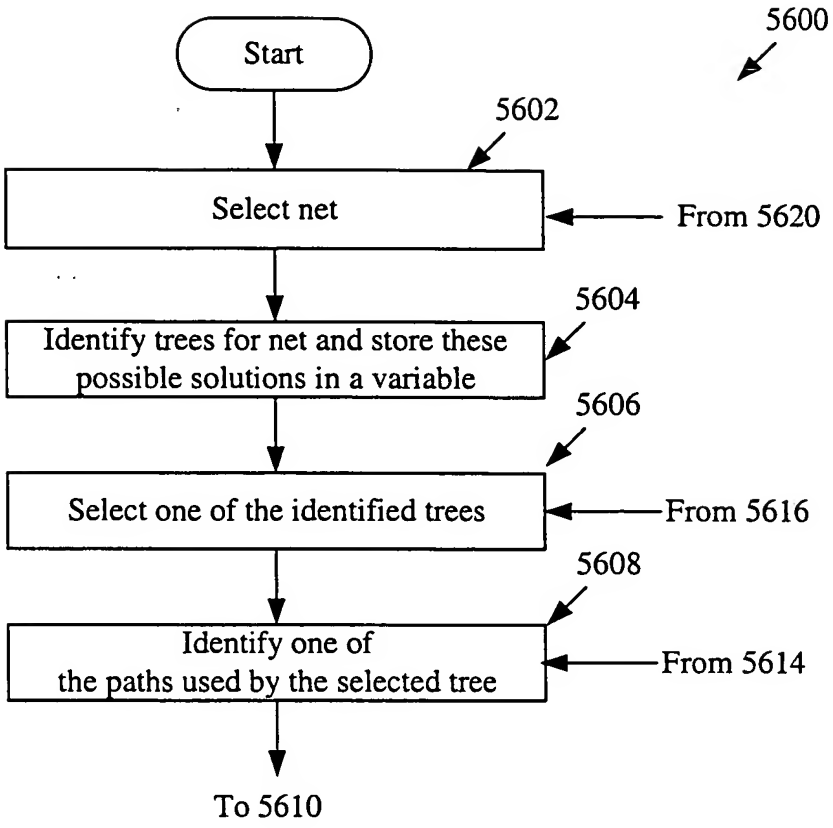

Figure 56A

Figure 56: *Figure 56A* / *Figure 56B* / *Figure 56C* / *Figure 56D*

Docket No. 50502-0004

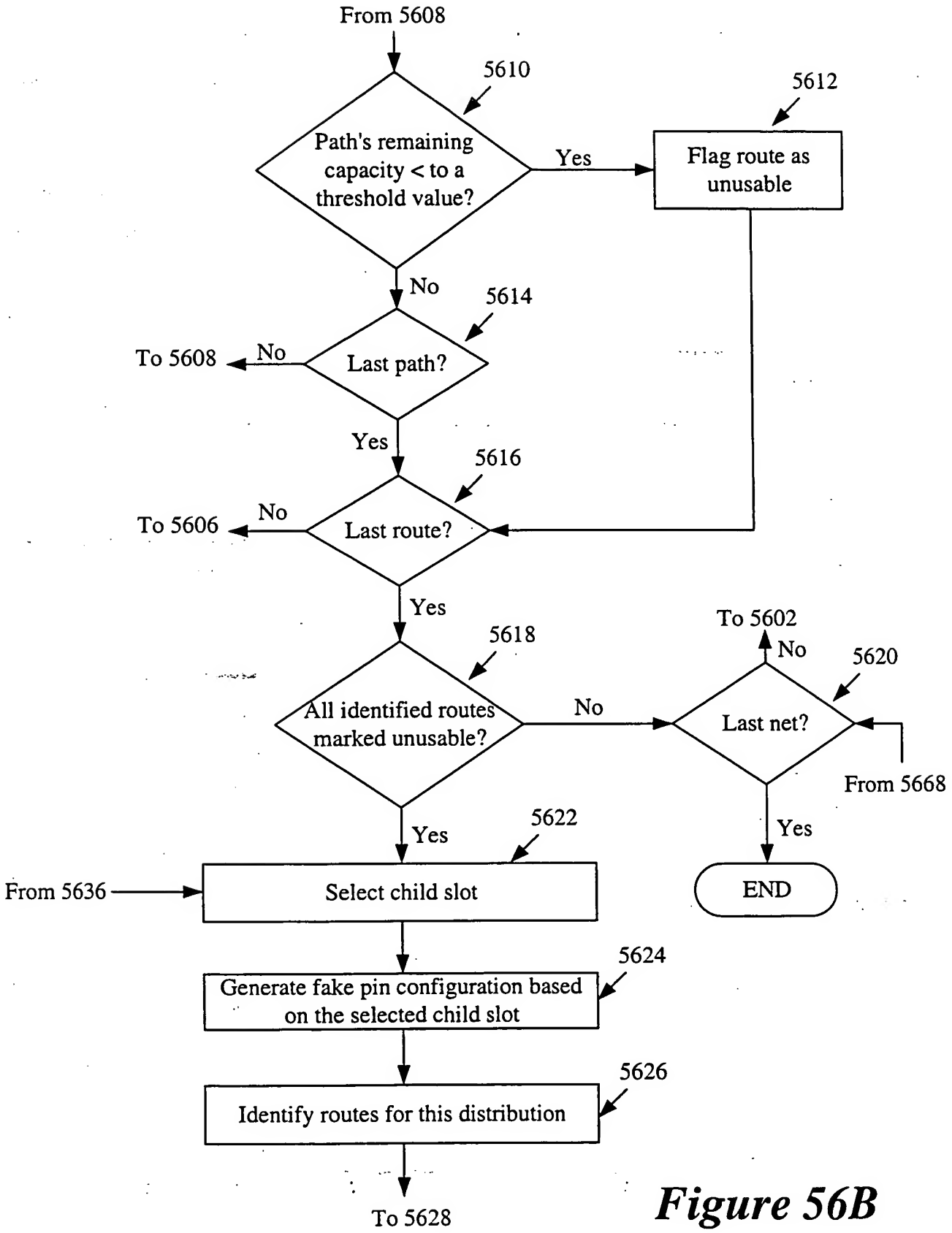

Figure 56B

20040049302

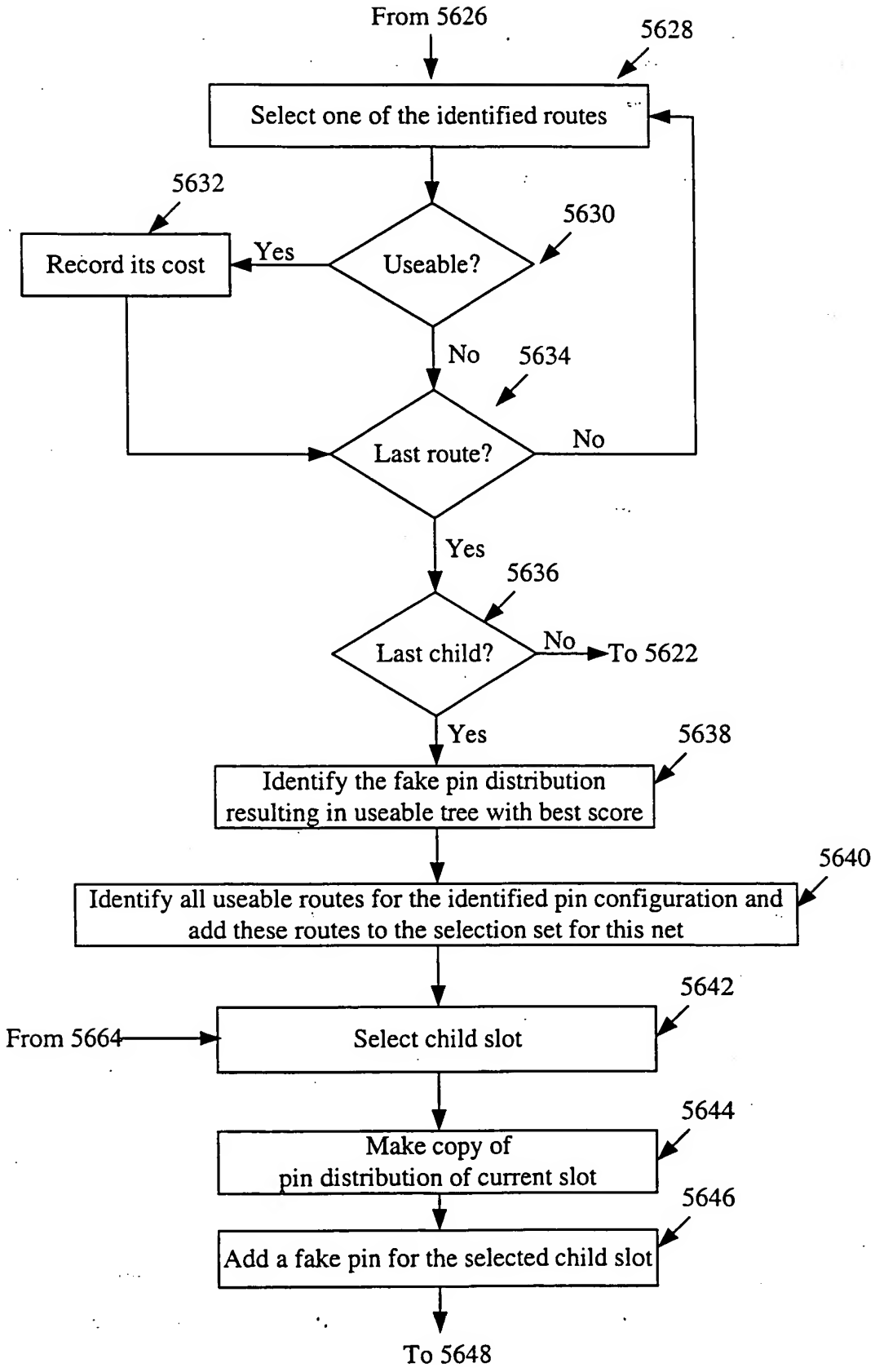

Figure 56C

Docket No. 95-1-0004

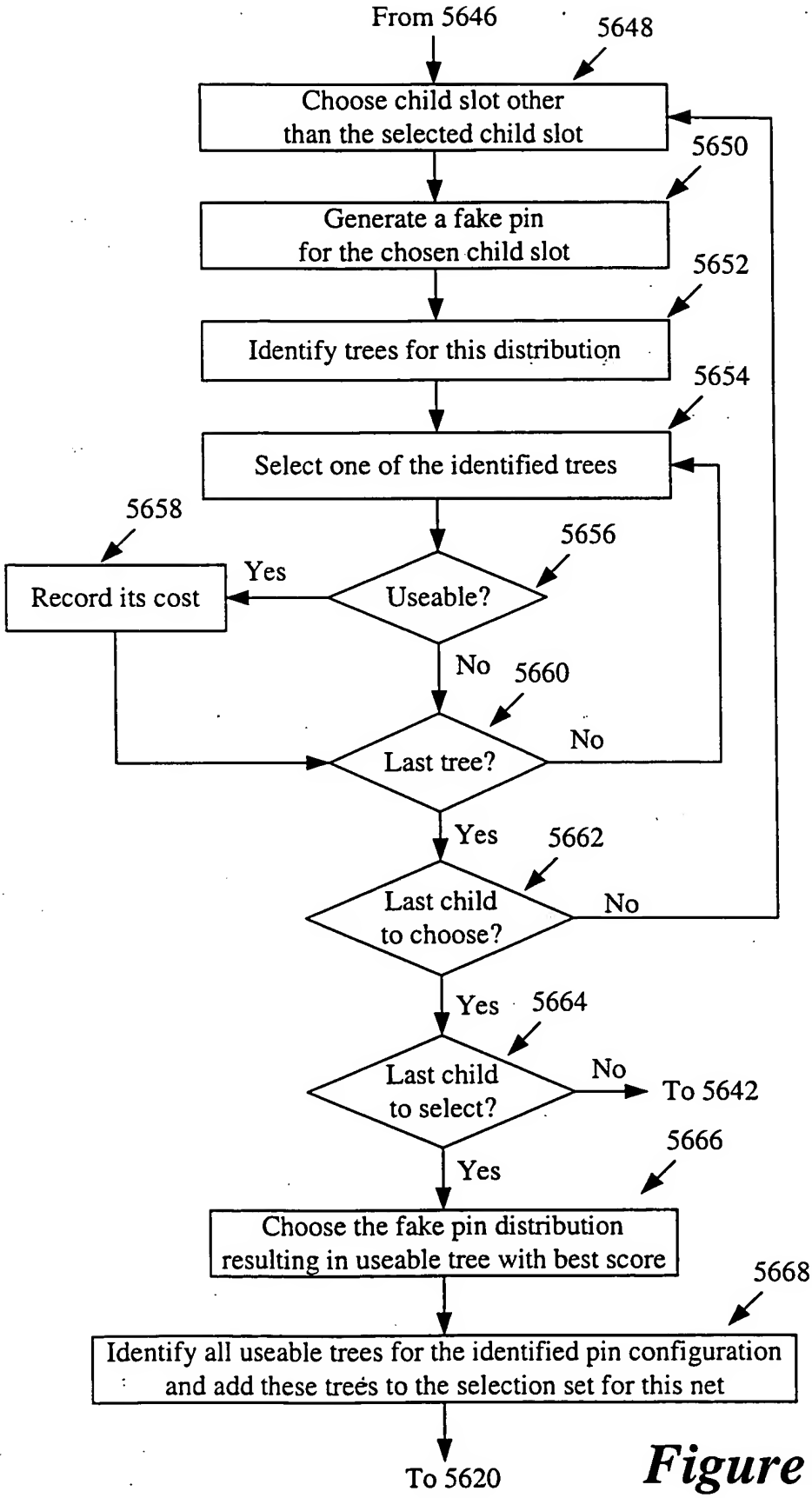

Figure 56D

20050405.0704

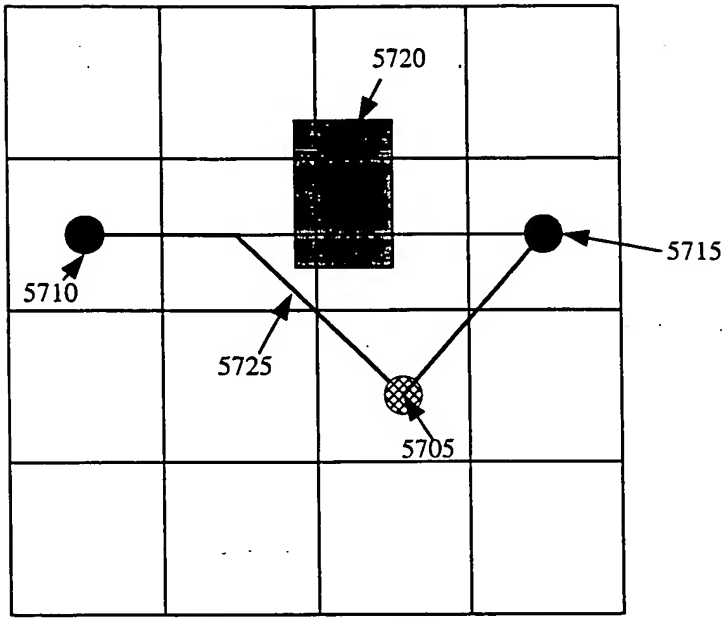

Figure 57

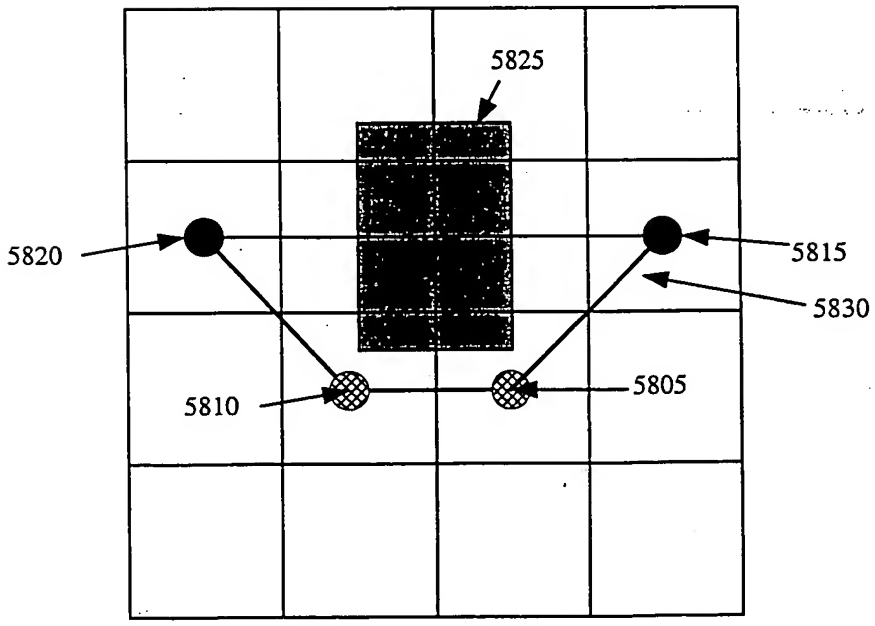

Figure 58

20050704.0950.0176 - 05/09/2005 - 15:04:04

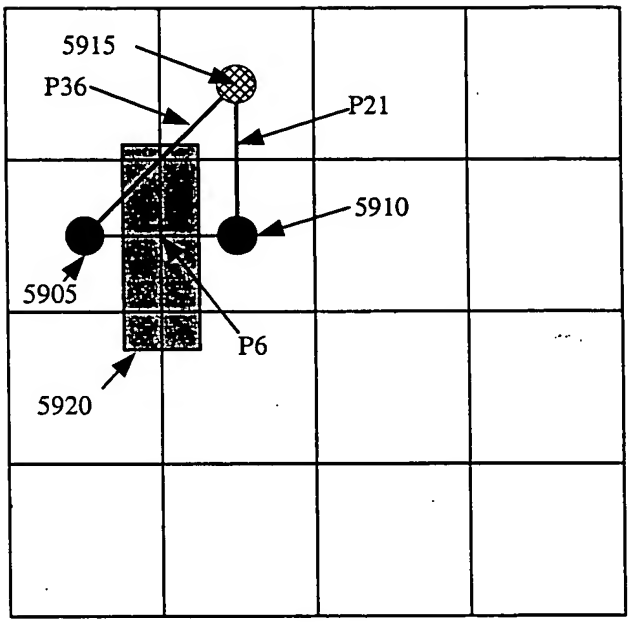

Figure 59

20050150502 - 20050404

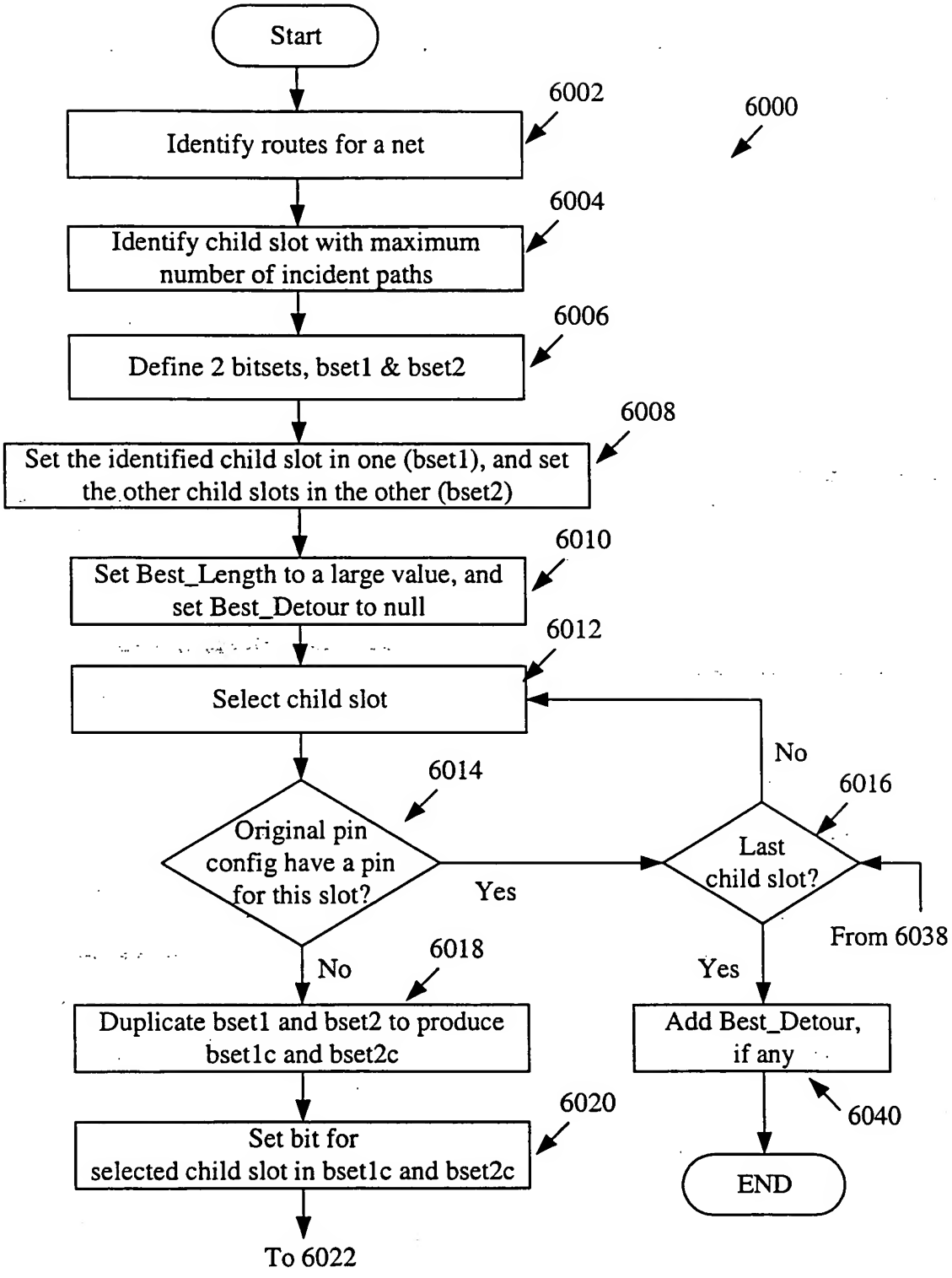

Figure 60A

Figure 60: *Figure 60A* / *Figure 60B*

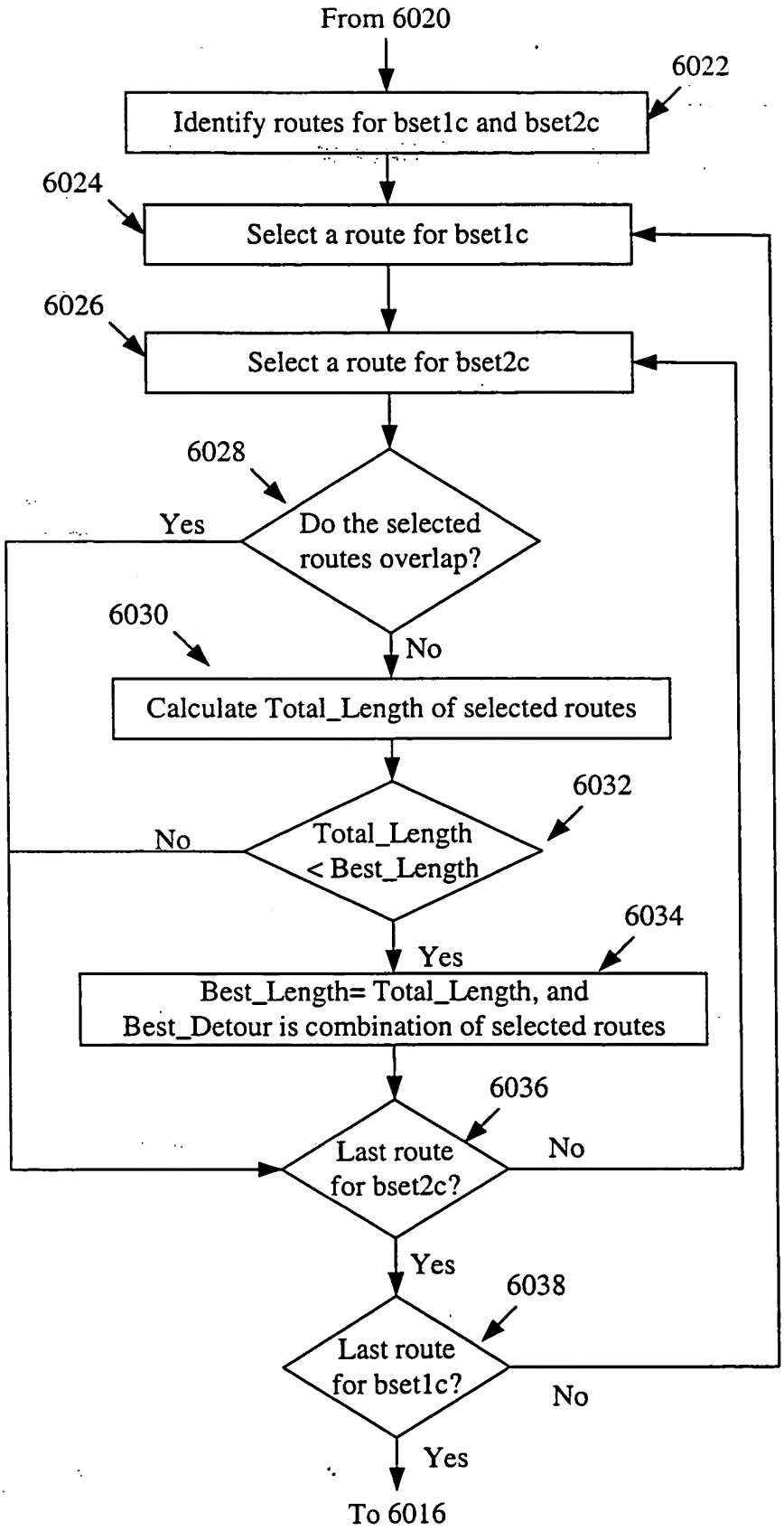
From 6020
6022
Identify routes for bset1c and bset2c
6024
Select a route for bset1c
6026
Select a route for bset2c
6028
Do the selected routes overlap?
Yes
No
6030
Calculate Total_Length of selected routes
6032
Total_Length < Best_Length
No
Yes
6034
Best_Length= Total_Length, and Best_Detour is combination of selected routes
6036
Last route for bset2c?
No
Yes
6038
Last route for bset1c?
No
Yes
To 6016

Figure 60B

20070 E960n00T

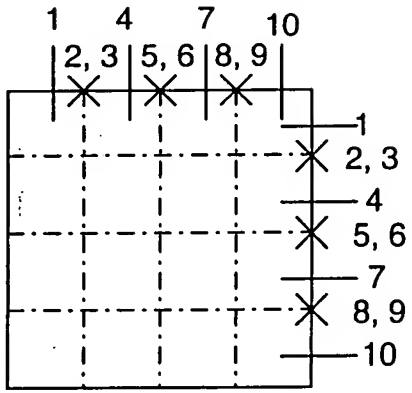

Figure 61

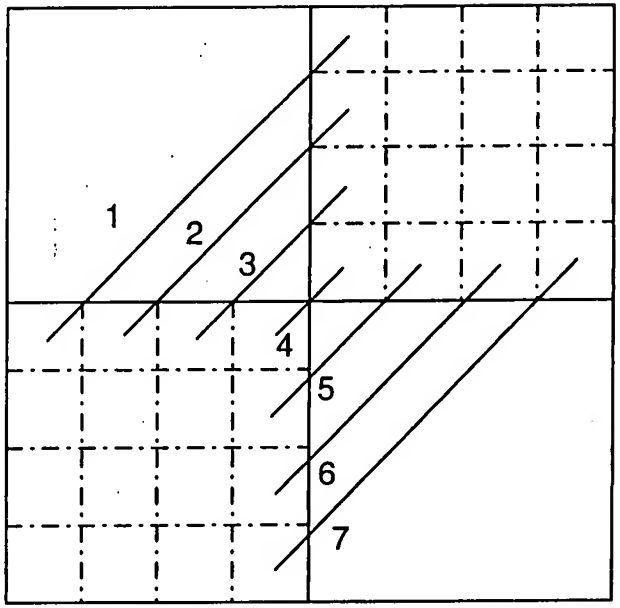

Figure 62

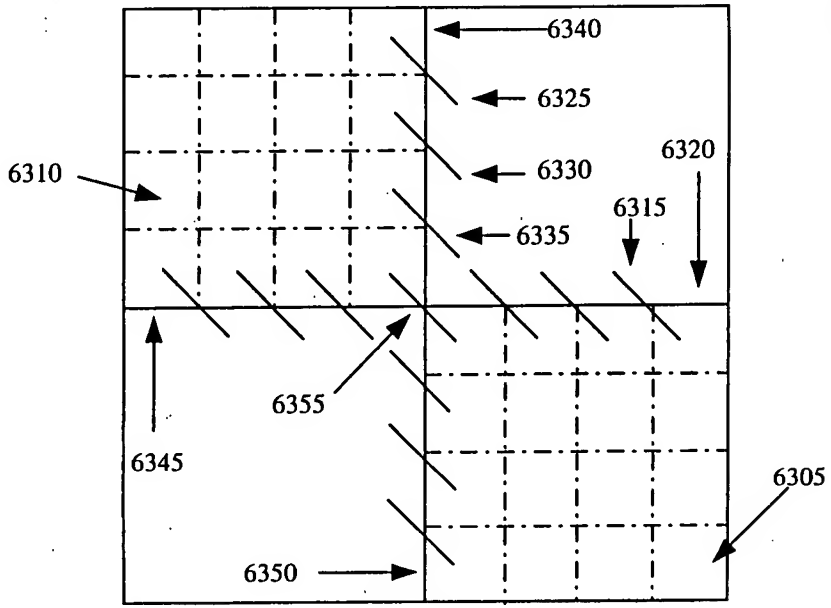

Figure 63

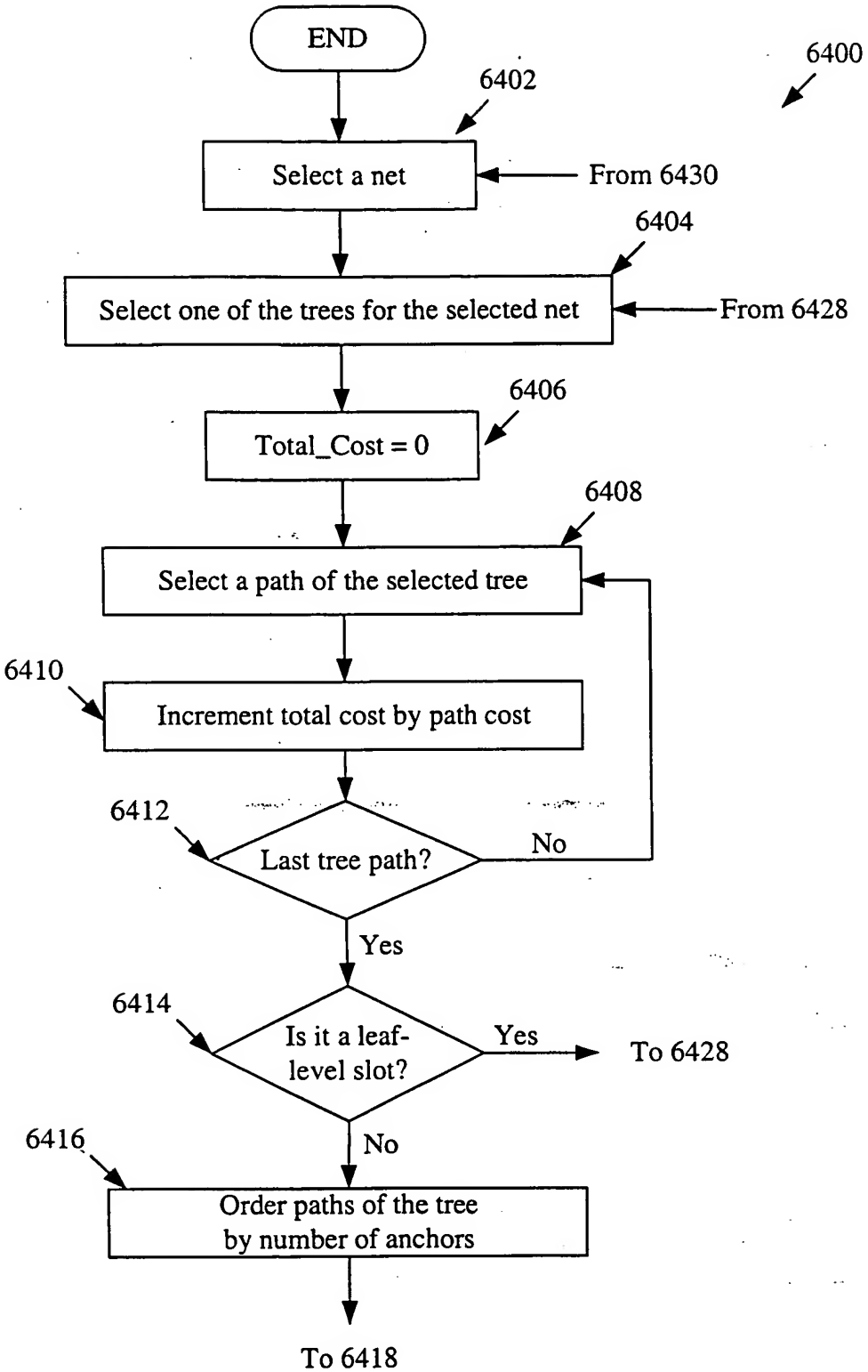

Figure 64A

Figure 64: Figure 64A / Figure 64B

Express Mail No. EV 045 945 325 US

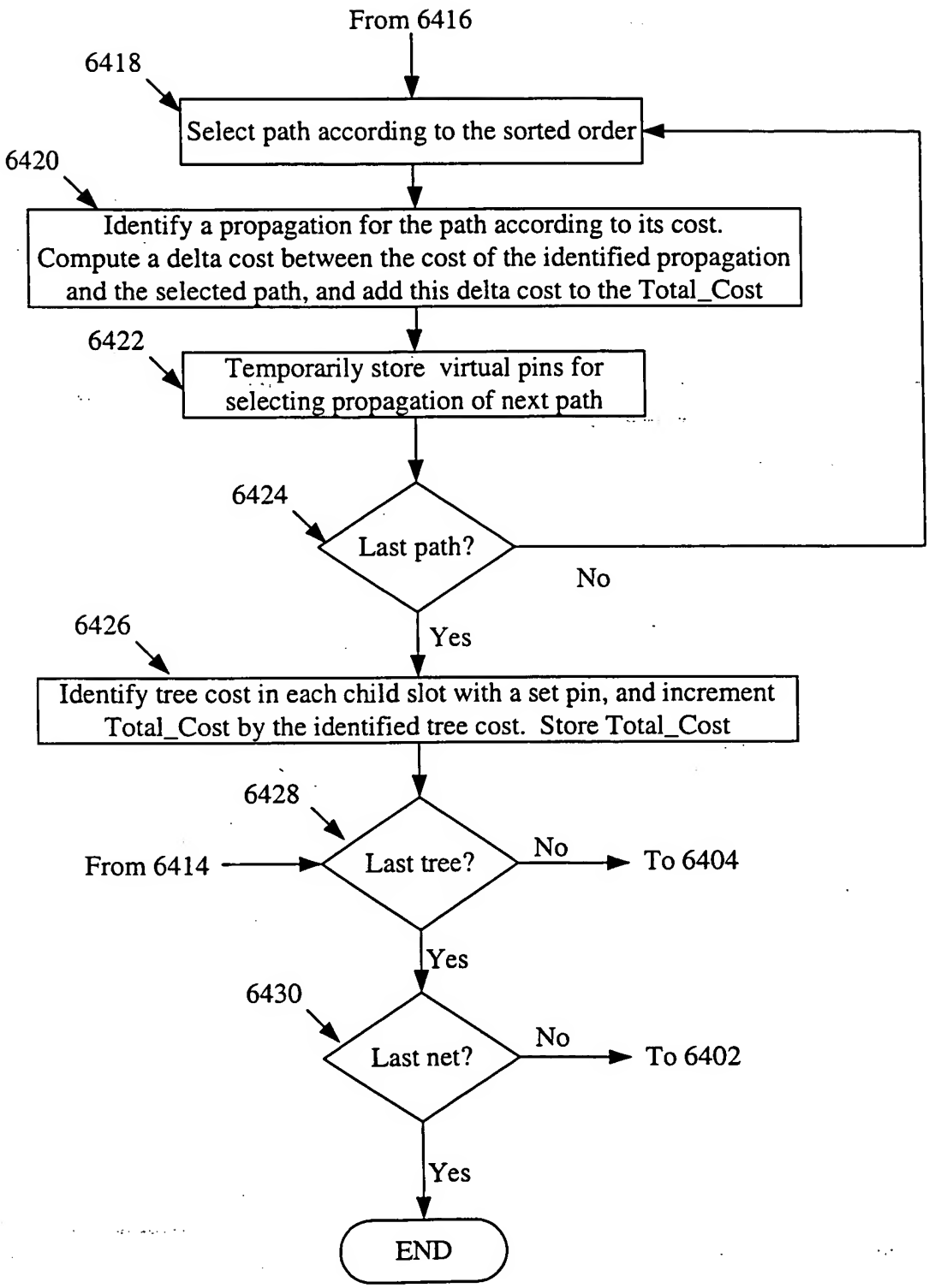

Figure 64B

20501D" E960h00f

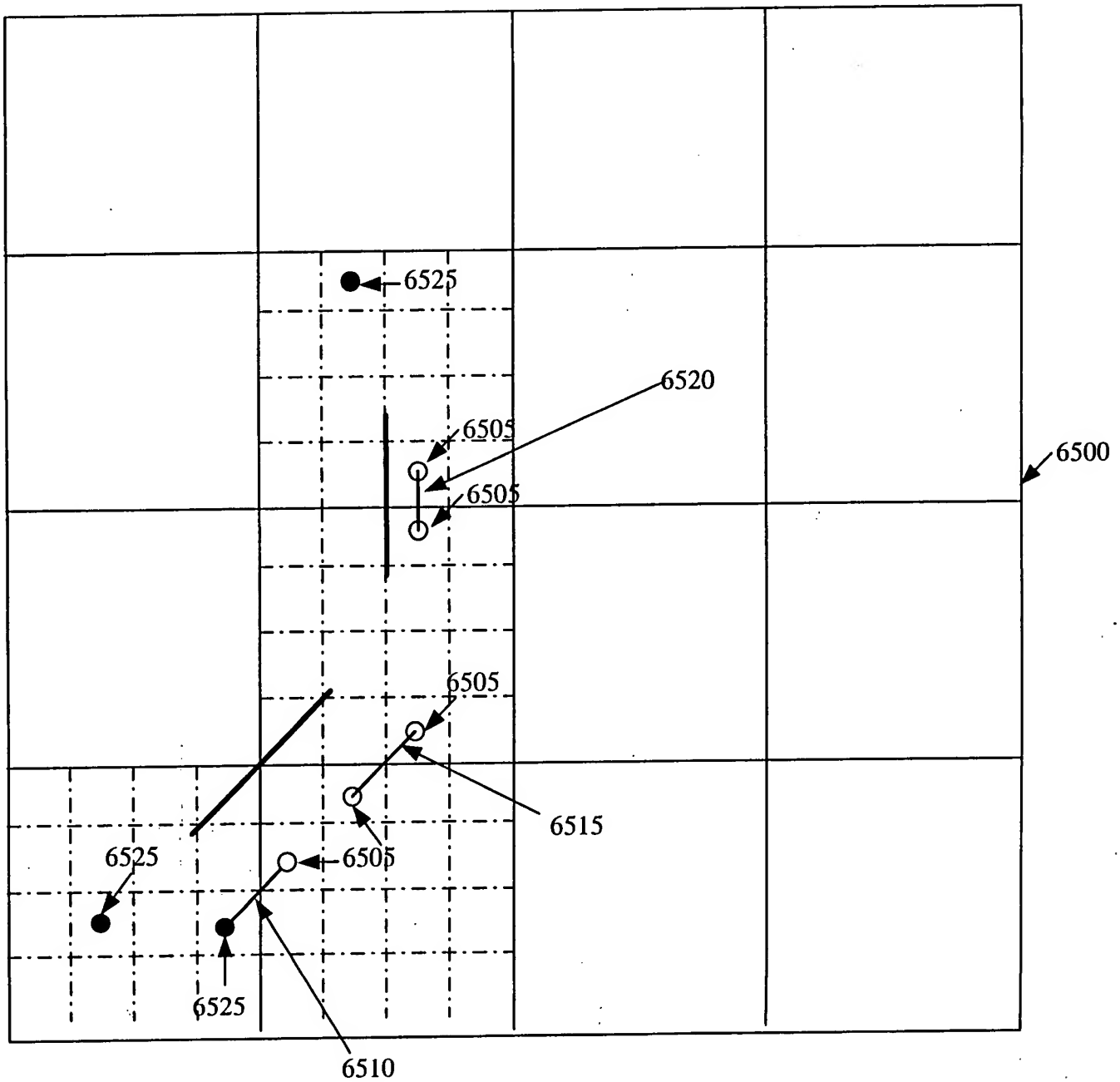

Figure 65

20070 "E960+00T

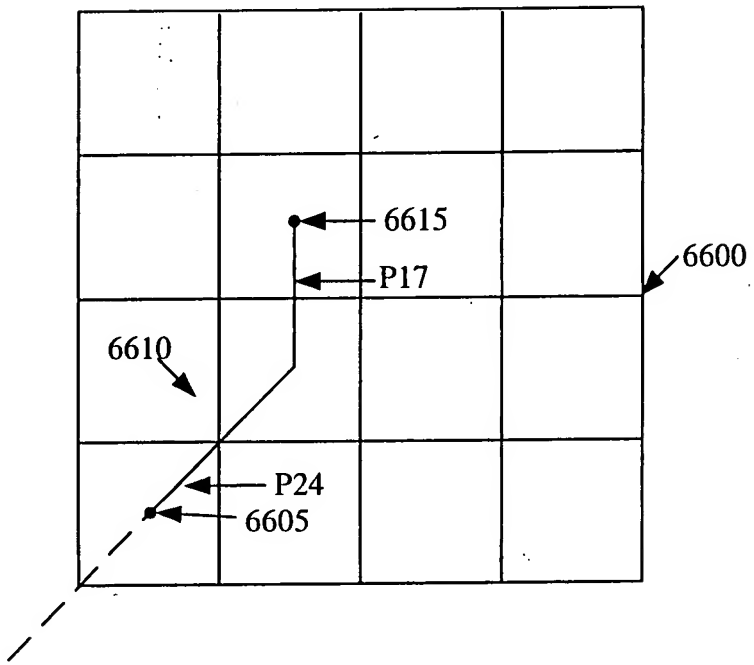

Figure 66

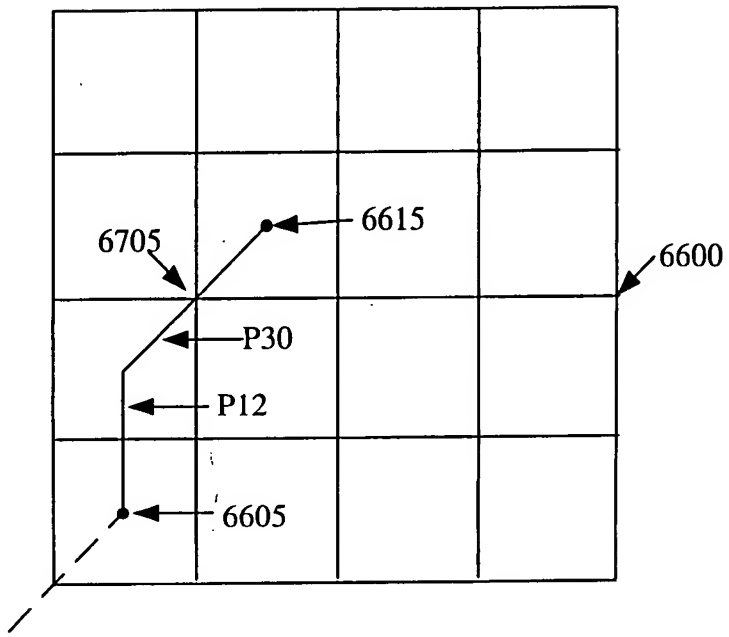

Figure 67

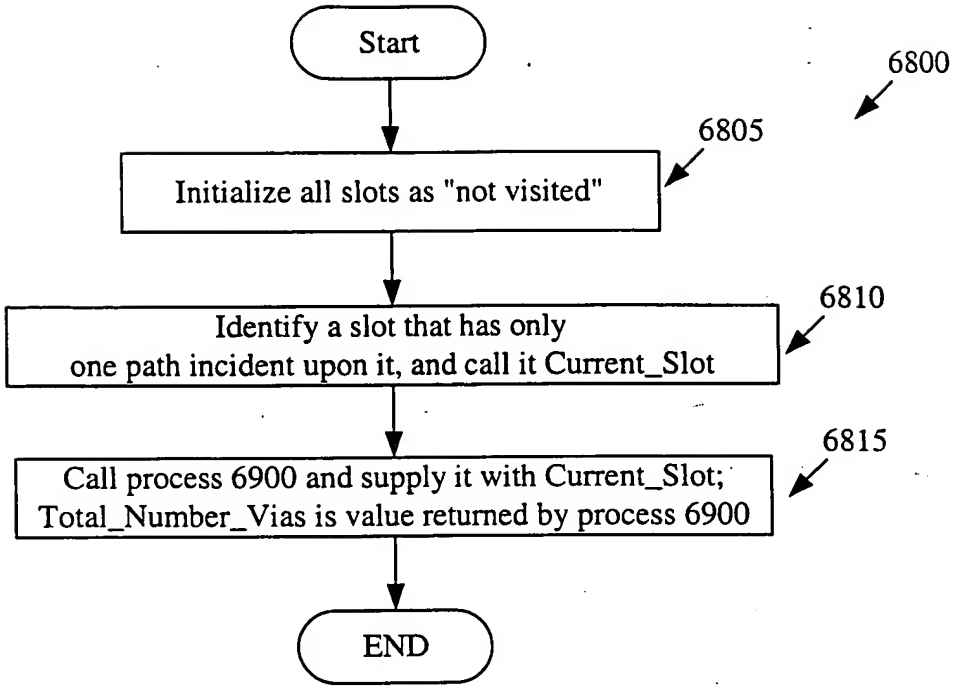

Figure 68

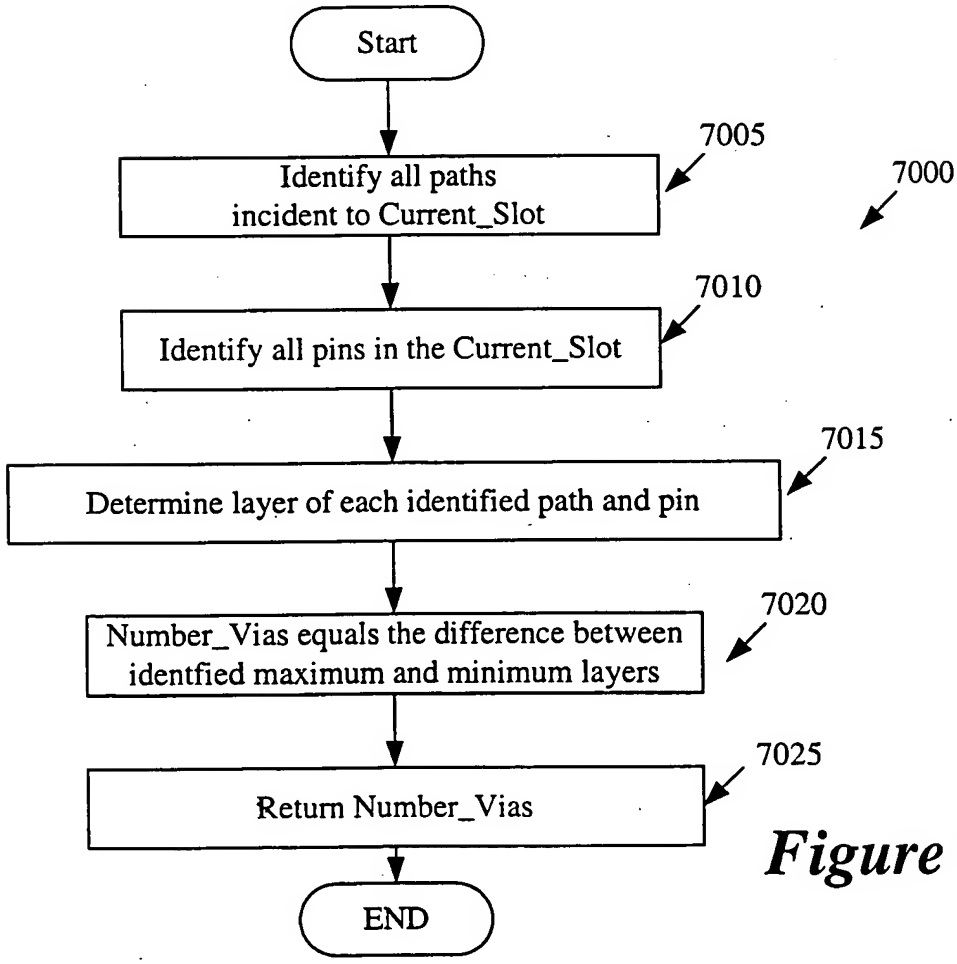

Figure 70

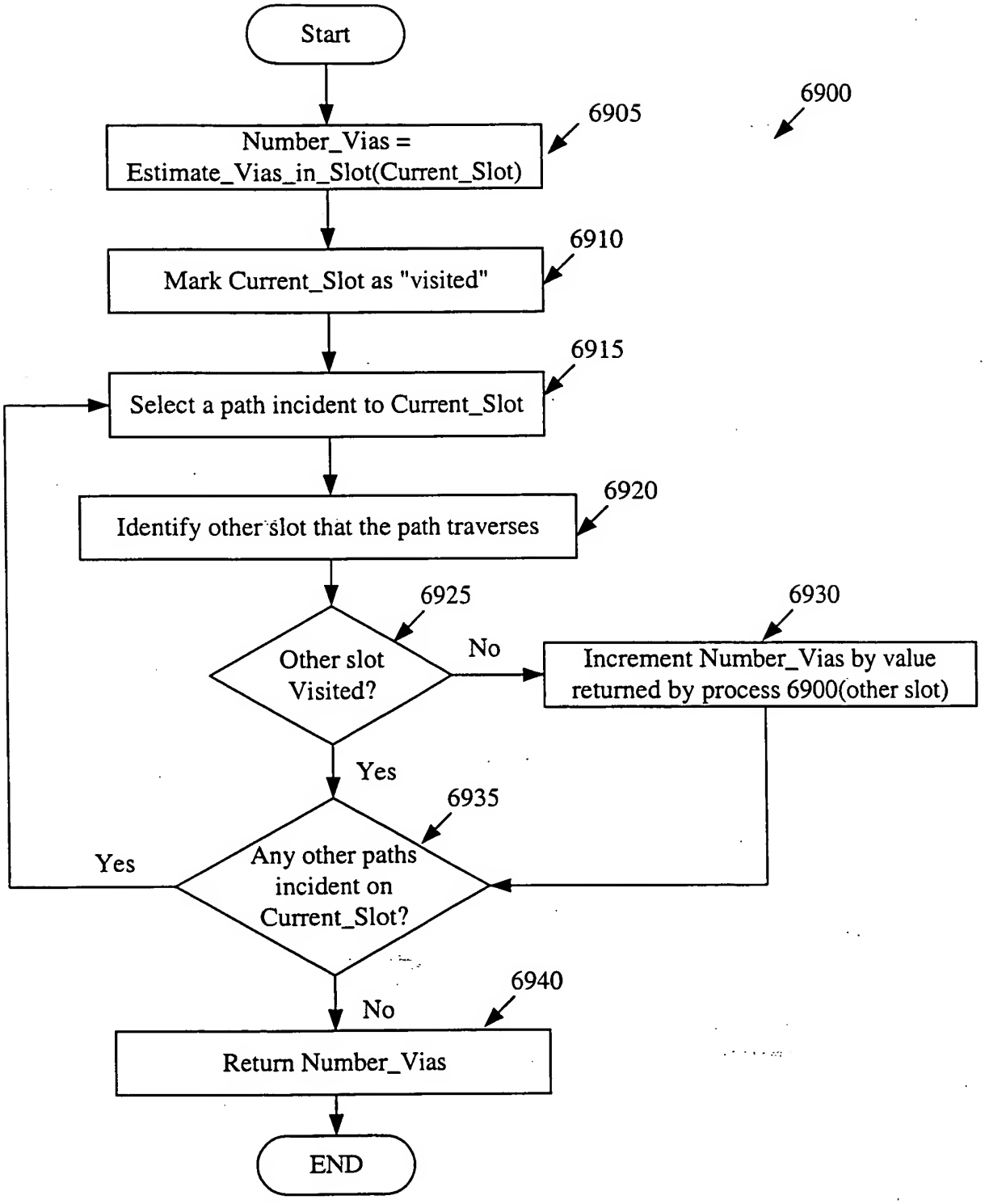
Start
6900
6905
Number_Vias =
Estimate_Vias_in_Slot(Current_Slot)
6910
Mark Current_Slot as "visited"
6915
Select a path incident to Current_Slot
6920
Identify other slot that the path traverses
6925
Other slot
Visited?
No
6930
Increment Number_Vias by value
returned by process 6900(other slot)
Yes
6935
Yes
Any other paths
incident on
Current_Slot?
No
6940
Return Number_Vias
END

Figure 69

20501O" E960h00t

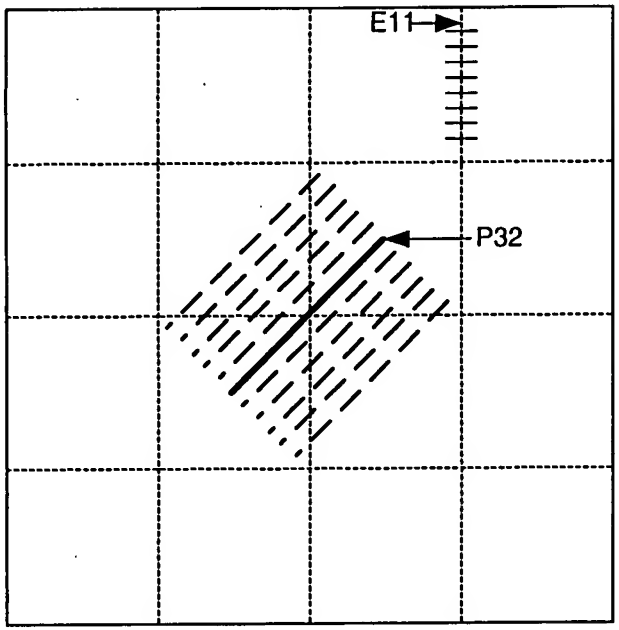

Figure 71

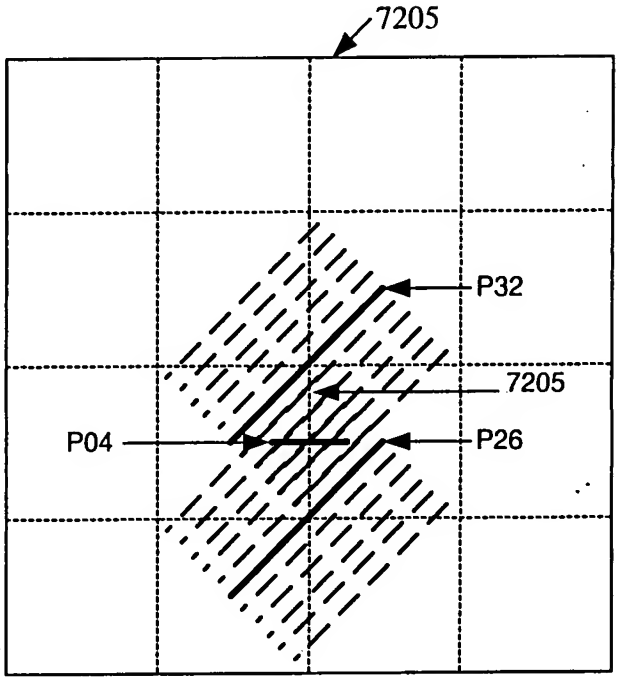

Figure 72

20070 E960h00t

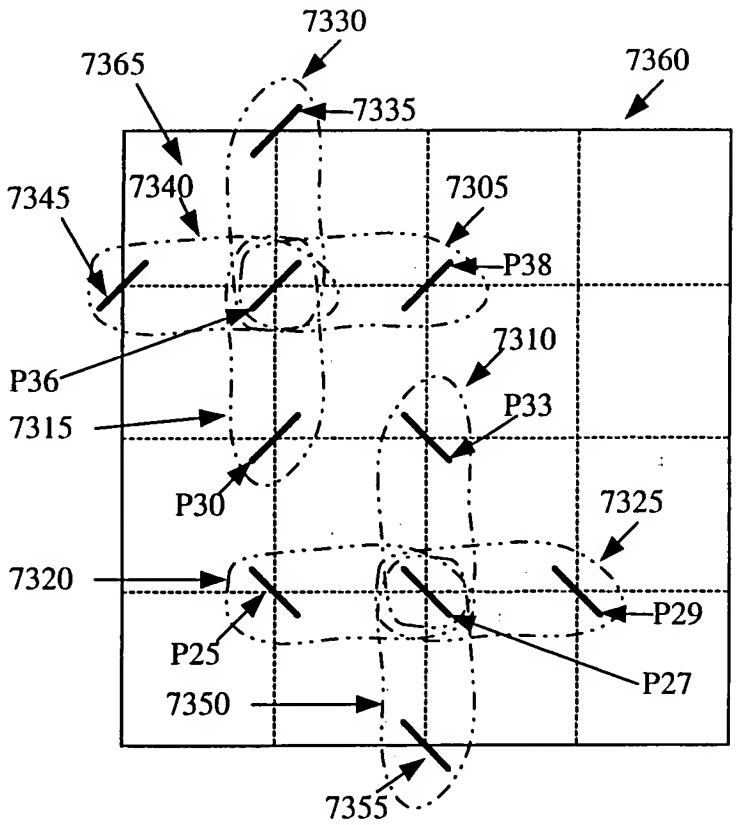

Figure 73

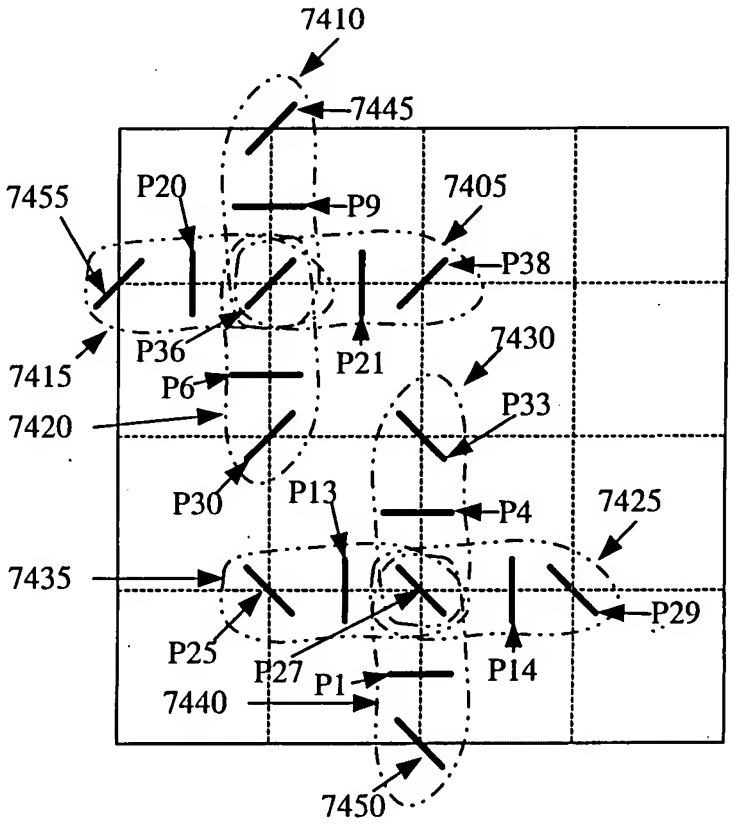

Figure 74

20501D"E960h00T

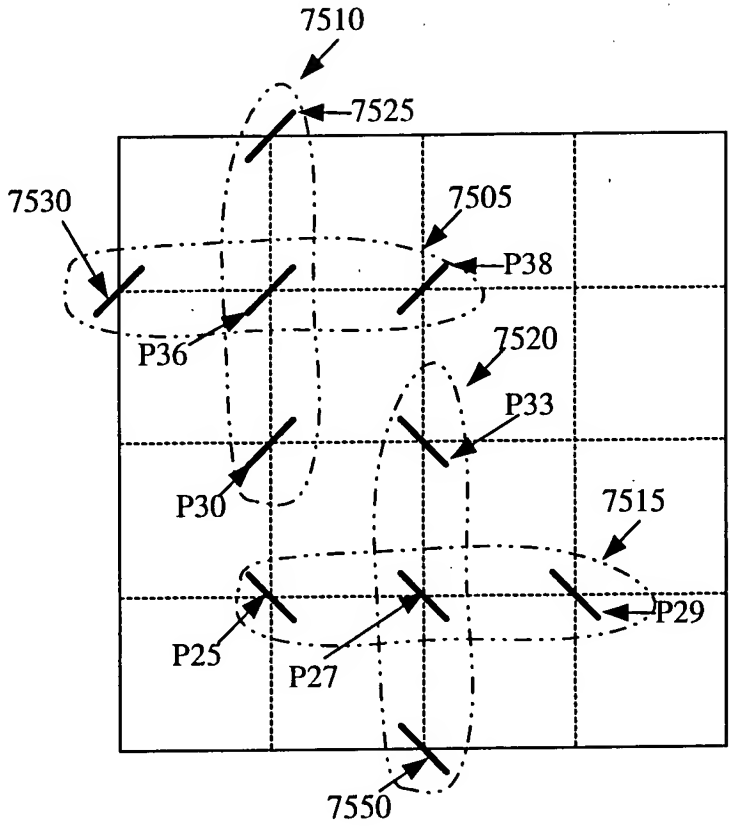

Figure 75

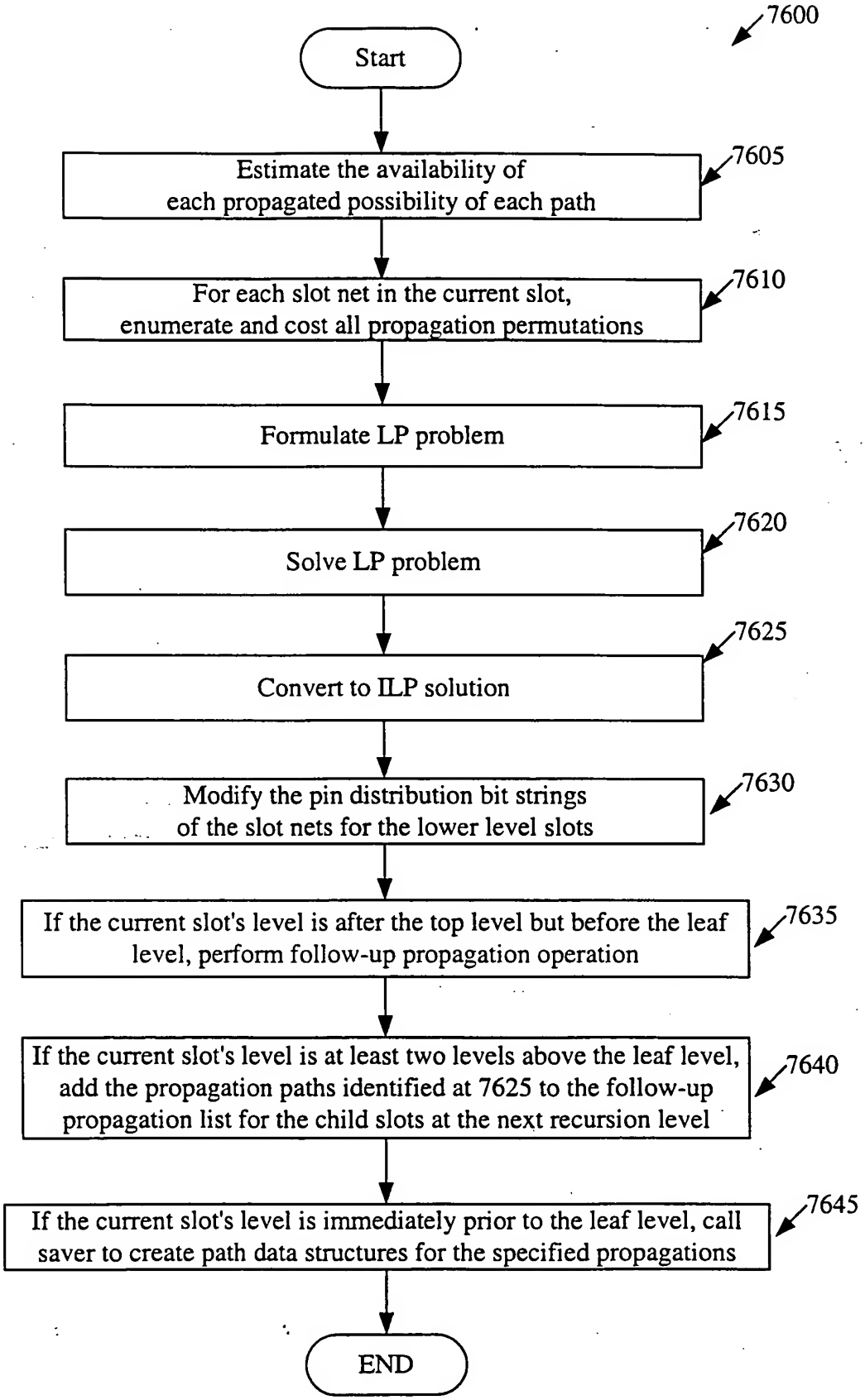
7600
Start
Estimate the availability of each propagated possibility of each path
7605
For each slot net in the current slot, enumerate and cost all propagation permutations
7610
Formulate LP problem
7615
Solve LP problem
7620
Convert to ILP solution
7625
Modify the pin distribution bit strings of the slot nets for the lower level slots
7630
If the current slot's level is after the top level but before the leaf level, perform follow-up propagation operation
7635
If the current slot's level is at least two levels above the leaf level, add the propagation paths identified at 7625 to the follow-up propagation list for the child slots at the next recursion level
7640
If the current slot's level is immediately prior to the leaf level, call saver to create path data structures for the specified propagations
7645
END

Figure 76

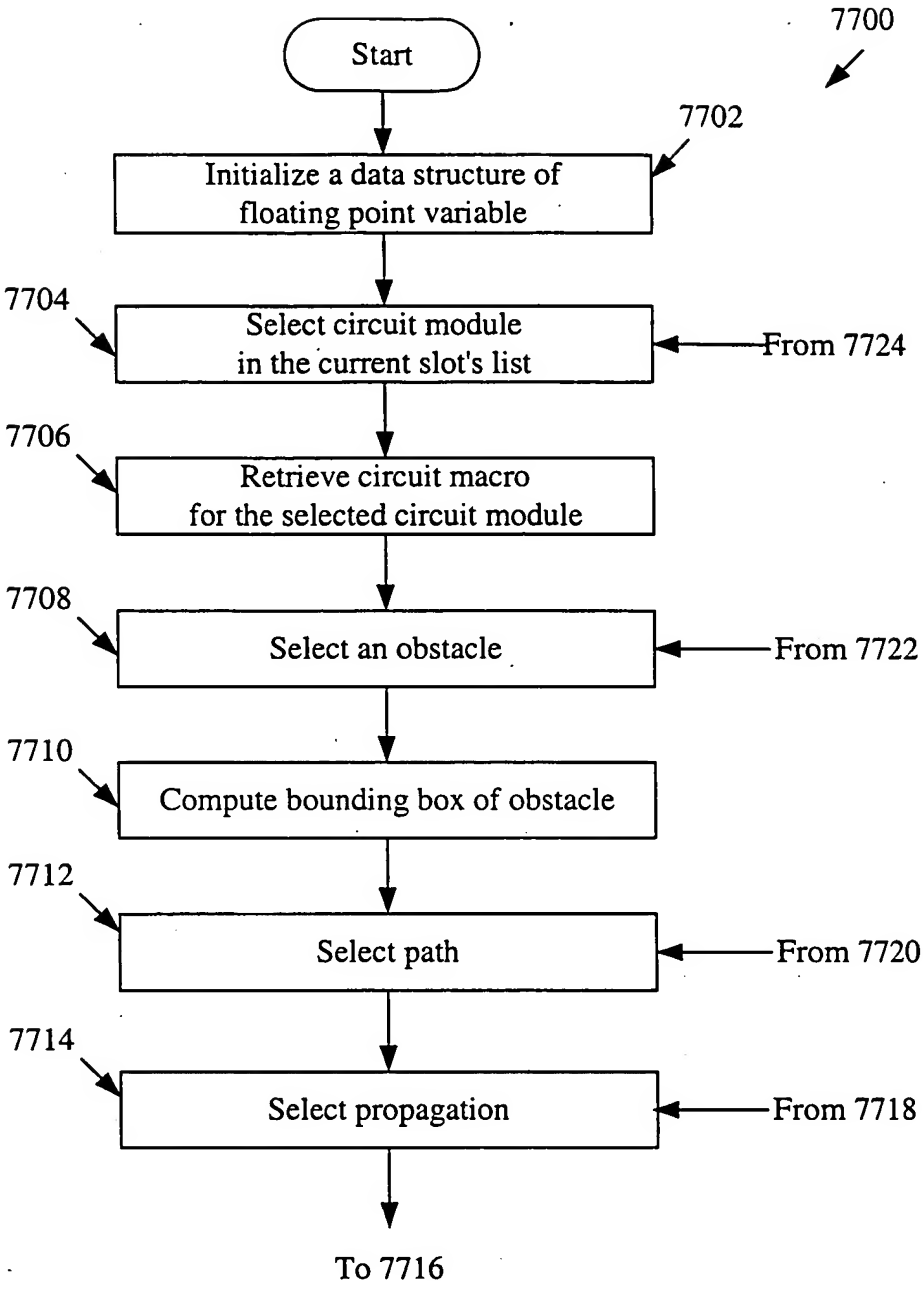

Figure 77A

Figure 77: Figure 77A / Figure 77B

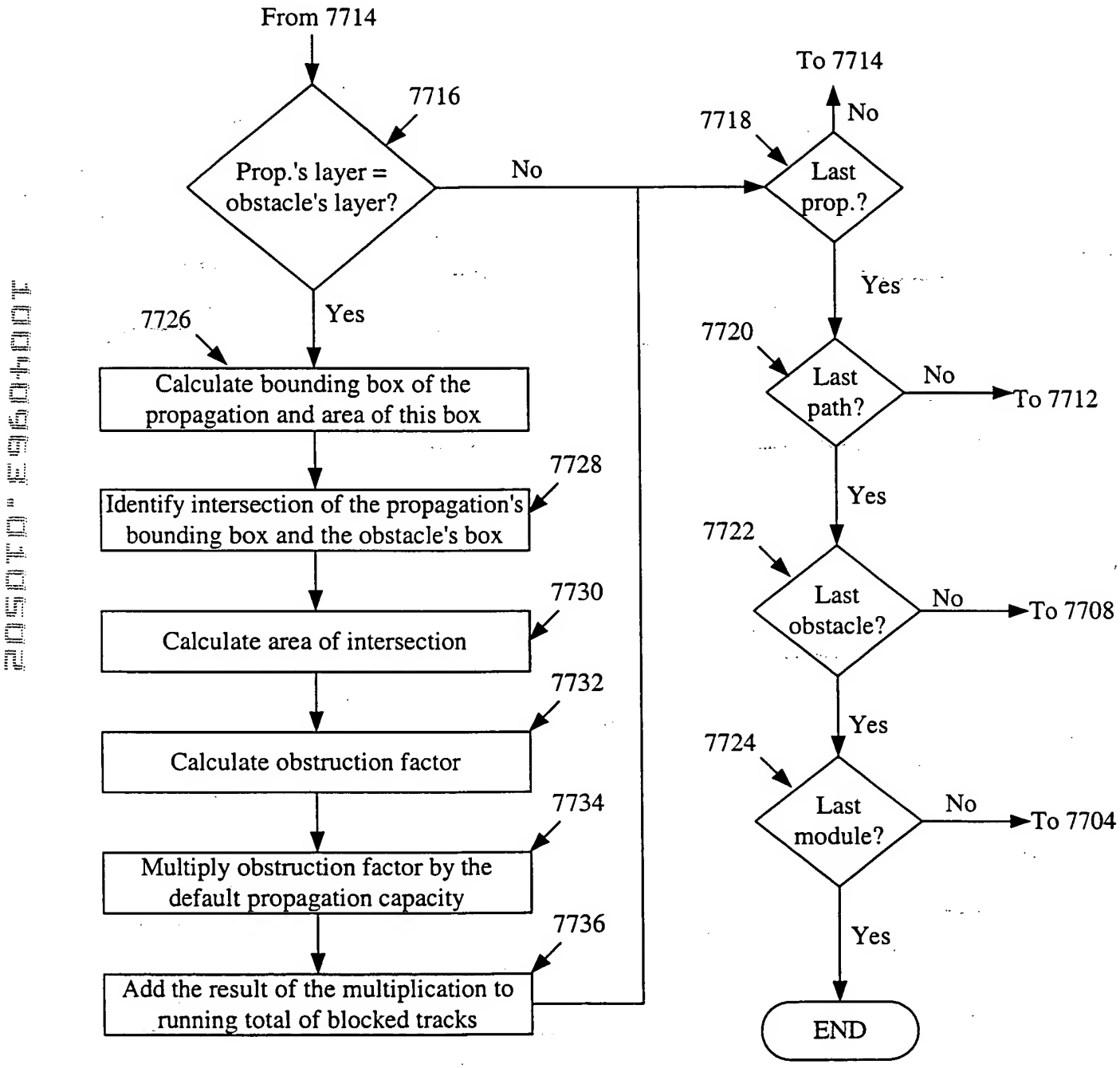
From 7714
7716
Prop.'s layer = obstacle's layer?
No
Yes
7726
Calculate bounding box of the propagation and area of this box
7728
Identify intersection of the propagation's bounding box and the obstacle's box
7730
Calculate area of intersection
7732
Calculate obstruction factor
7734
Multiply obstruction factor by the default propagation capacity
7736
Add the result of the multiplication to running total of blocked tracks
To 7714
No
7718
Last prop.?
Yes
7720
Last path?
No
To 7712
Yes
7722
Last obstacle?
No
To 7708
Yes
7724
Last module?
No
To 7704
Yes
END

Figure 77B

Attorney Docket No. 2005-0502

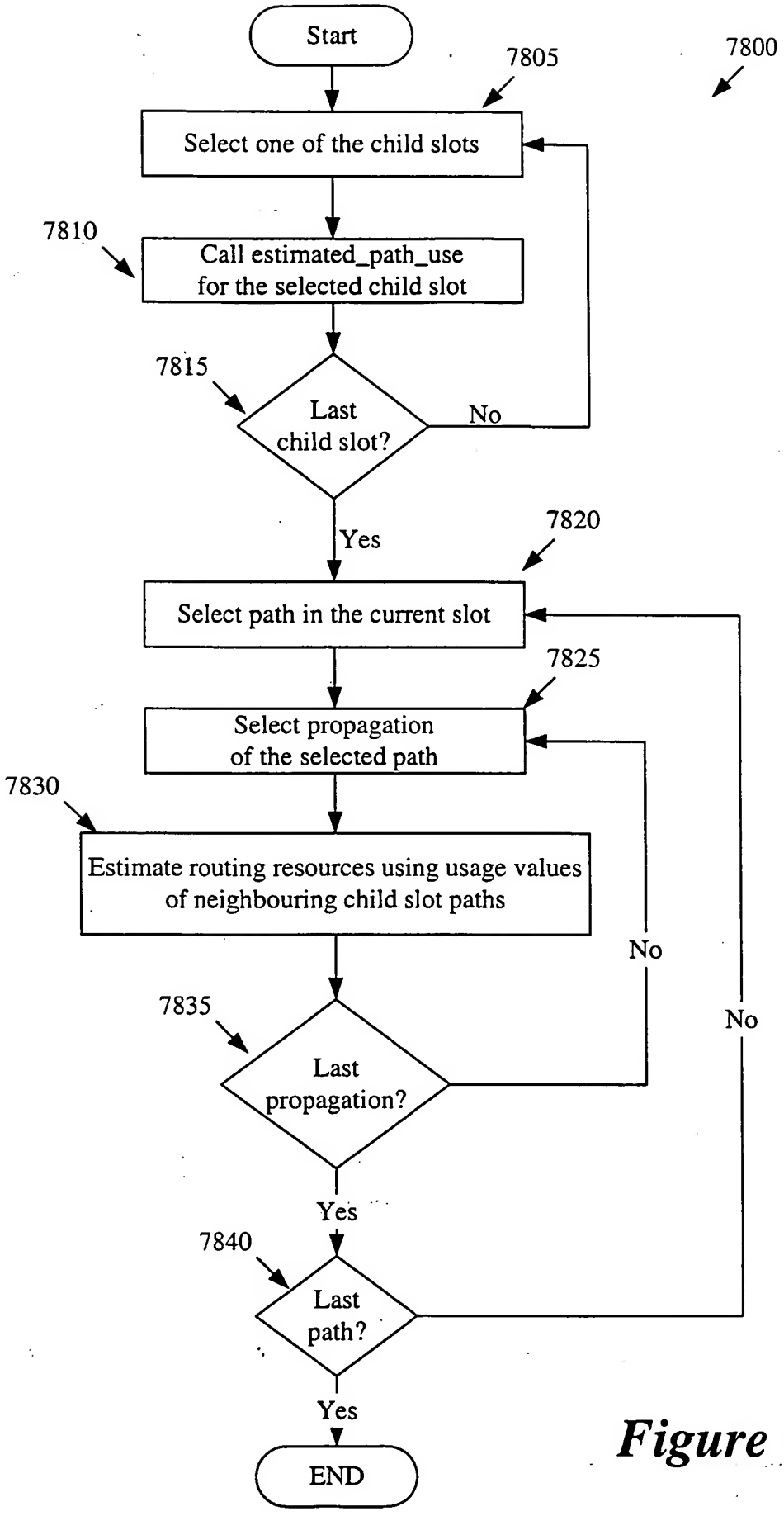

Figure 78

Application No. 09/684,095 Docket No. 3659-0102

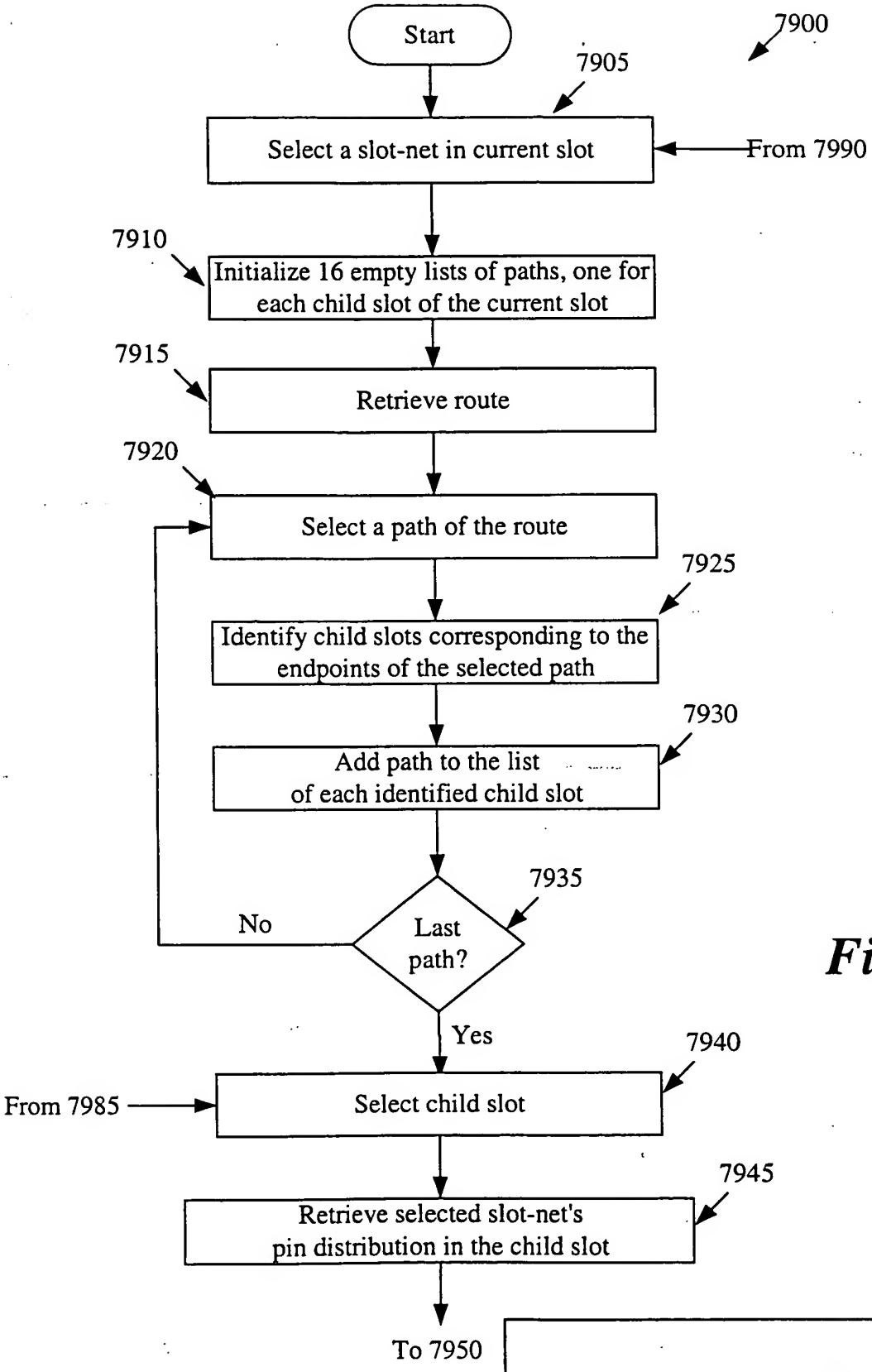

Figure 79A

Figure 79: *Figure 79A* / *Figure 79B*

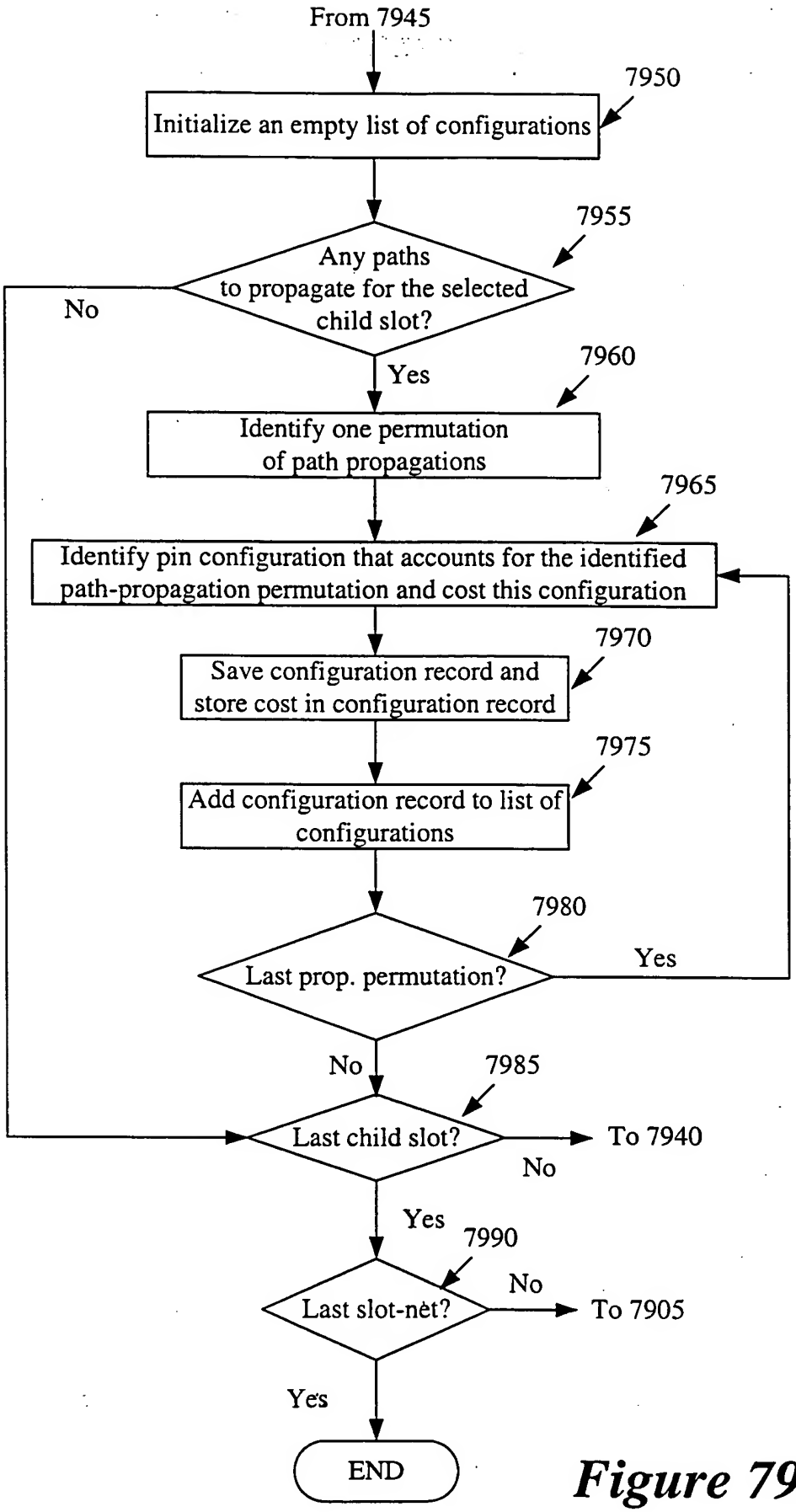
From 7945
7950
Initialize an empty list of configurations
7955
Any paths to propagate for the selected child slot?
No
Yes
7960
Identify one permutation of path propagations
7965
Identify pin configuration that accounts for the identified path-propagation permutation and cost this configuration
7970
Save configuration record and store cost in configuration record
7975
Add configuration record to list of configurations
7980
Last prop. permutation?
Yes
No
7985
Last child slot?
To 7940
No
Yes
7990
Last slot-net?
No
To 7905
Yes
END

Figure 79B

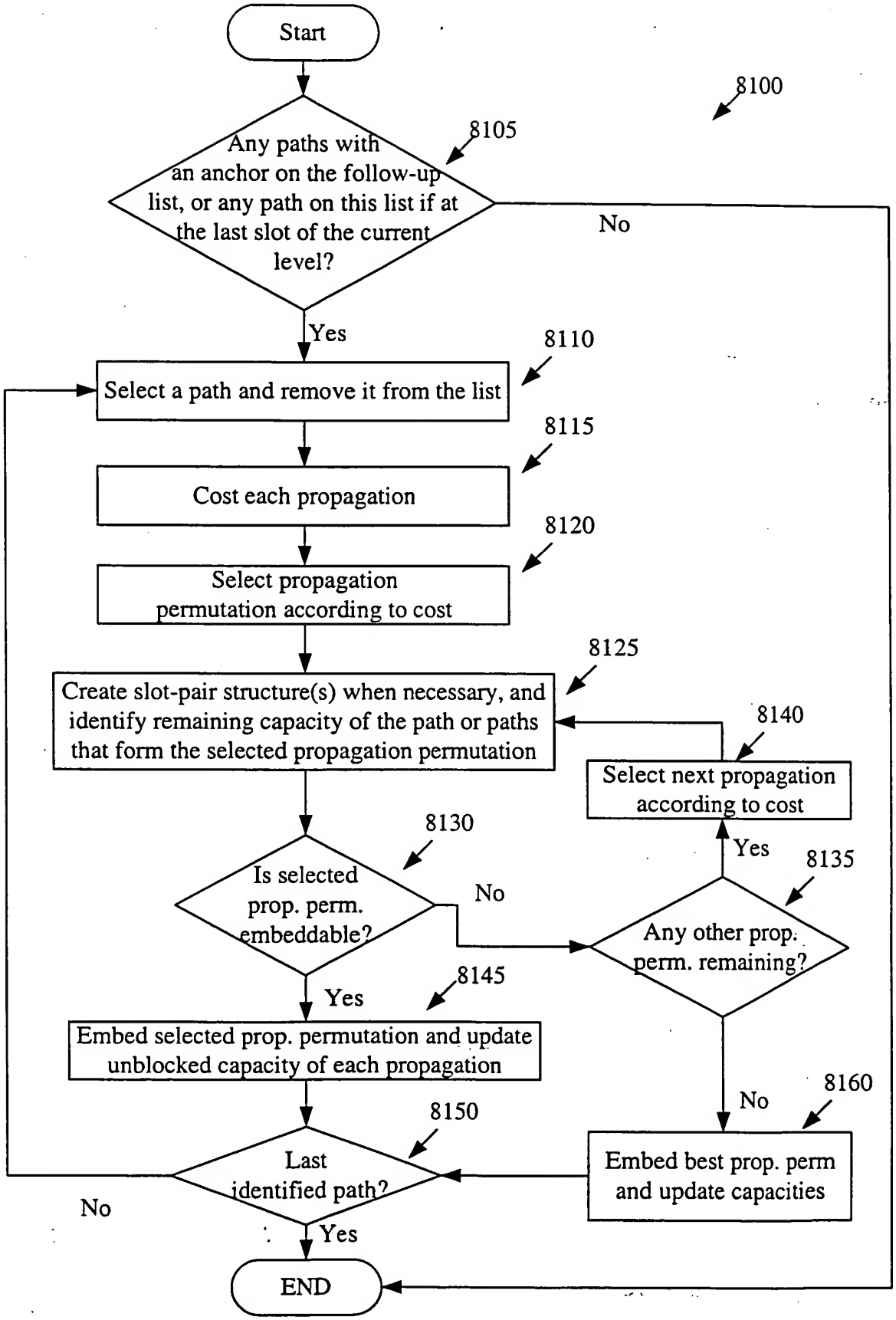
Start
8100
Any paths with an anchor on the follow-up list, or any path on this list if at the last slot of the current level?
8105
No
Yes
8110
Select a path and remove it from the list
8115
Cost each propagation
8120
Select propagation permutation according to cost
8125
Create slot-pair structure(s) when necessary, and identify remaining capacity of the path or paths that form the selected propagation permutation
8140
Select next propagation according to cost
Yes
8130
Is selected prop. perm. embeddable?
No
8135
Any other prop. perm. remaining?
Yes
8145
Embed selected prop. permutation and update unblocked capacity of each propagation
No
8160
8150
Last identified path?
Embed best prop. perm and update capacities
No
Yes
END

Figure 81

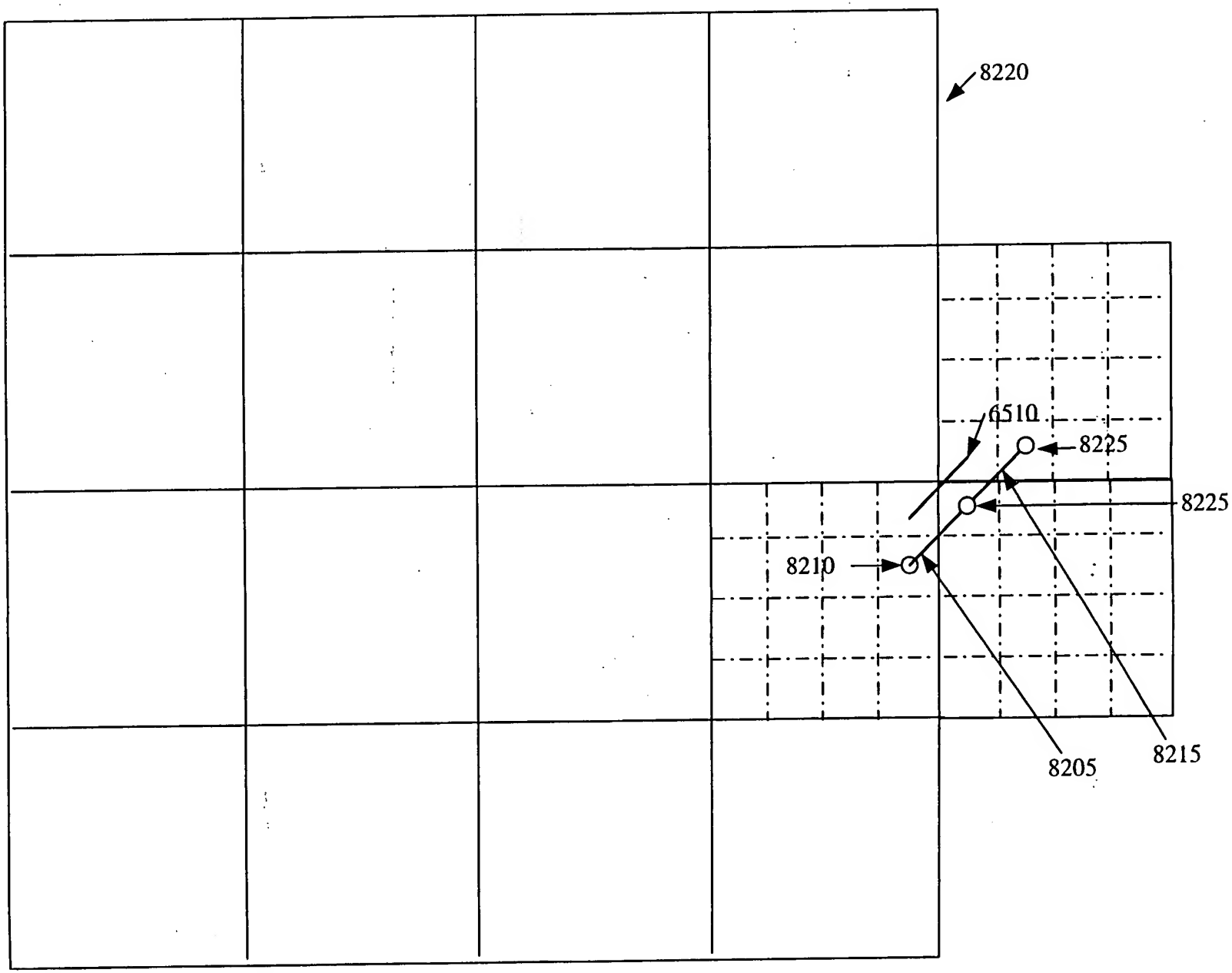

Figure 82

Attorney Docket No. 2005P04096US

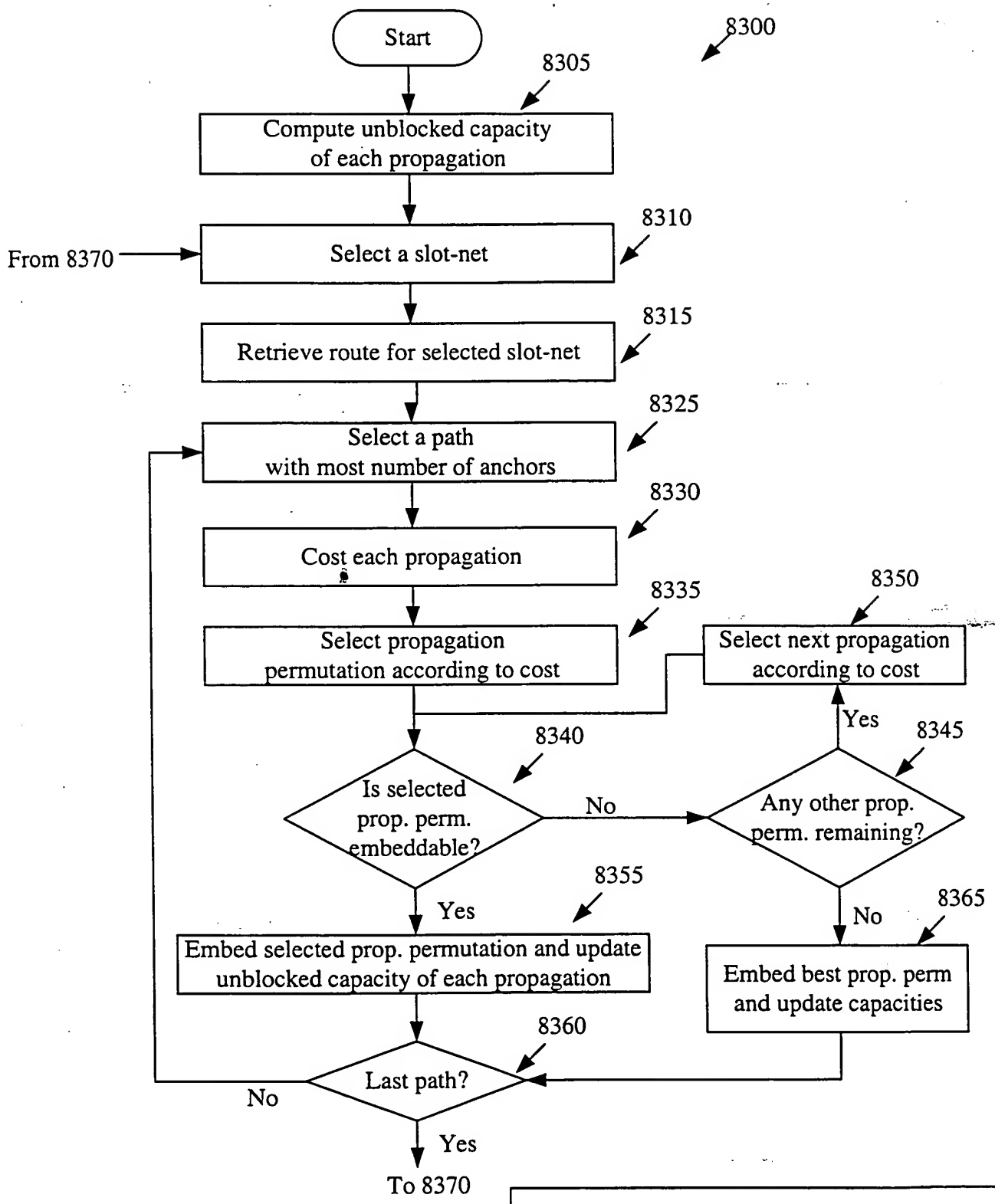

Figure 83A

Figure 83: *Figure 83A* / *Figure 83B*

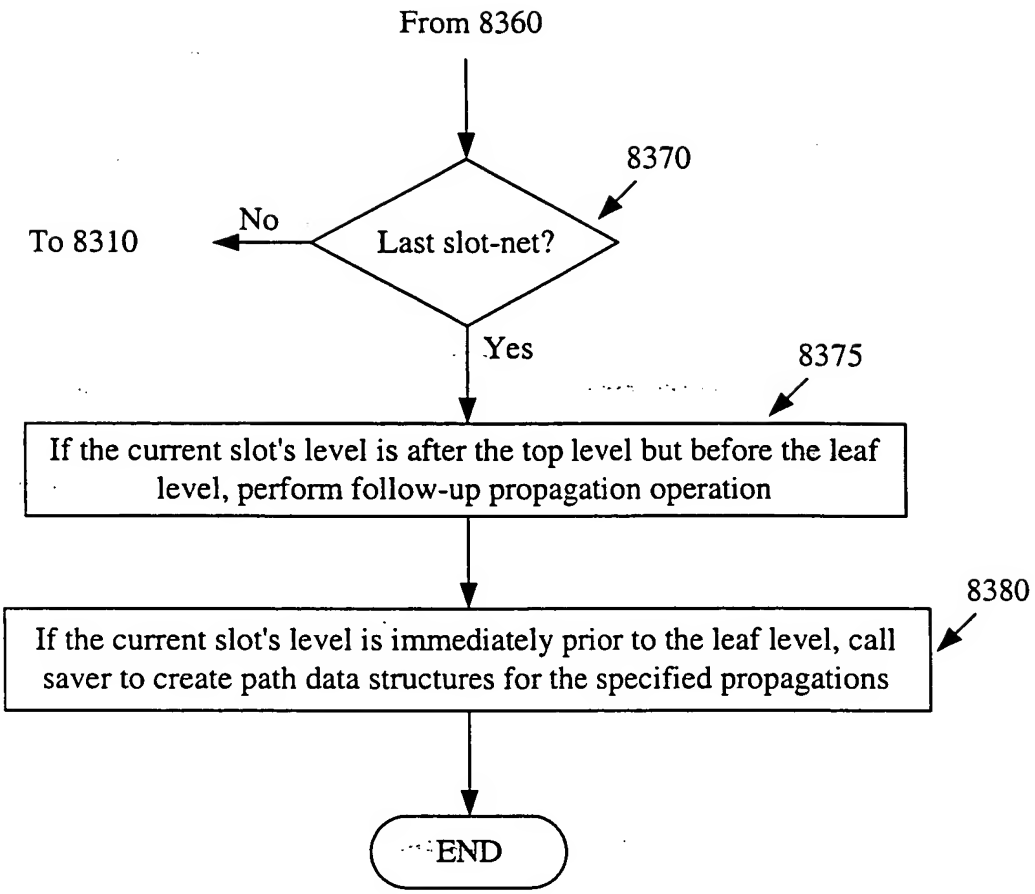
From 8360
8370
No
To 8310
Last slot-net?
Yes
8375
If the current slot's level is after the top level but before the leaf level, perform follow-up propagation operation
8380
If the current slot's level is immediately prior to the leaf level, call saver to create path data structures for the specified propagations
END

Figure 83B

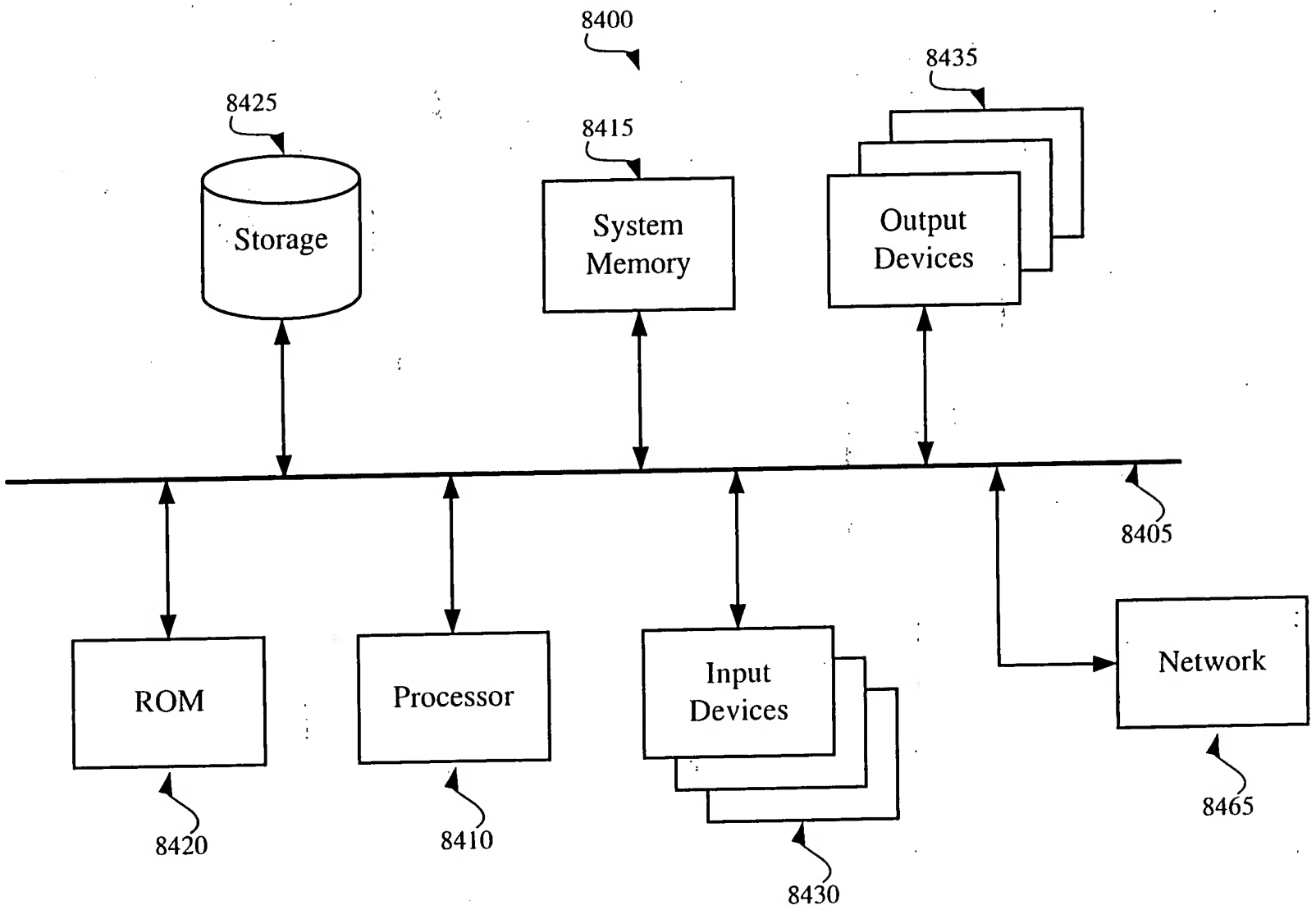
8400
8435
8425
8415
Storage
System Memory
Output Devices
8405
ROM
Processor
Input Devices
Network
8420
8410
8430
8465

Figure 84